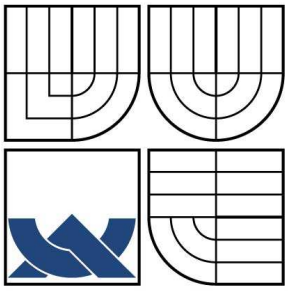
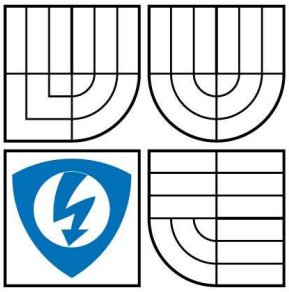


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

ŘEŠENÍ BEZPEČNOSTI V IMS

IMS SECURITY SOLUTIONS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

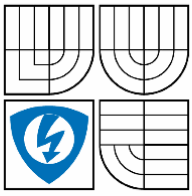
AUTOR PRÁCE — Bc. TOMÁŠ PORUBSKÝ

AUTHOR

VEDOUCÍ PRÁCE — Ing. TOMÁŠ MÁCHA

SUPERVISOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Tomáš Porubský	***ID:***	83331
Ročník:	2	***Akademický rok:***	2008/2009

NÁZEV TÉMATU:

Řešení bezpečnosti v IMS

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je prostudovat a popsat technologii IMS (IP Multimedia Subsystem). Zaměřte se na základní bezpečnostní procedury v IMS (autentizace, autorizace) a na typy rozhraní (Cx, Dx, Sh, atd.). Proveďte realizaci IMS sítě v systému Open IMS Core. Na základě získaných poznatků realizujte laboratorní úlohu. Proveďte analýzu přenášených zpráv.

DOPORUČENÁ LITERATURA:

[1] POIKSELKA, Miikka, MAYER, Gregor, KHARTABIL, Hisham. The IMS: IP Multimedia Concepts and Services. England : WILEY, 2006. 431 s. Second edition. ISBN 0-470-01906-9.
[2] CAMARILLO, Gonzalo, GACÍA-MARTÍN, Miguel A. The 3G IP Multimedia Subsystem (IMS). England : WILEY, 2006. 427 s. Second edition. ISBN 0-470-01818-6.

Termín zadání:	9.2.2009	***Termín odevzdání:***	26.5.2009

Vedoucí práce: Ing. Tomáš Mácha

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

V mé diplomové práci nejprve představuji obecnou architekturu sítě IMS (IP Multimedia Subsystem). Jde zejména o databázi účastníků HSS (Home Subscriber Server) a SLF (Subscription Locator Function), dále pak SIP servery CSCF (Call Session Control Functions), zpracovávající SIP signalizaci, poté aplikační servery AS (Application Server), vykonávající služby atd. Dále se zaměřuji na registraci účastníků v síti IMS s výpisem přenášených zpráv a popis jednotlivých rozhraní, které se v této síti používají. Mezi nejdůležitější zde patří rozhraní Gm, Mw, Cx, Dx a Sh. Následně zde popisuji problematiku bezpečnosti v IMS, která se dělí na zabezpečení přístupu a síťovou bezpečnost.

Poté přistupuji k samotné realizaci IMS sítě v open source systému Open IMS Core pod linuxovým operačním systémem. Zde popisuji průběh od samotné instalace tohoto systému, přes konfiguraci všech potřebných prvků sítě, až po samotnou komunikaci účastníku. V této části také provádím analýzu přenášené komunikace, jak při samotné registraci, tak i při následné komunikaci. V závěru jsem vytvořil laboratorní úlohu se zaměřením na tento systém Open IMS Core. Studenti se zde seznámí s architekturou a principem fungování sítí založených na technologii IMS, s jednotlivými prvky, nutnými pro provoz této sítě a jejich základním nastavením. Studenti si dále vyzkouší jednoduchou analýzu zachycené komunikace.

Klíčová slova: síť IMS, SIP protokol, bezpečnost, registrace, účastník, signalizace

ABSTRACT

In the first part of my master's thesis the network architecture of IMS (IP Multimedia Subsystem) is presented. The database of subscribers HSS (Home Subscriber Server) and SLF (Subscription Locator Function), as well as a SIP CSCF servers (Call Session Control Functions) process a SIP signalization and an AS application server performing services, etc. I focus on the registration of subscribers in the IMS network with a list of transmitted messages and description of each interface that is used in this network. The most important interfaces, which I described here, are Gm, Mw, Cx, Dx and Sh. Then I focused on security in IMS problems, which are divided into categories of access security and network security.

After that is the implementation of IMS network in an open source Open IMS Core System considered under the Linux operating system. Here is the problem description from the actual system installation, through the configuration of all necessary elements of the network to the communication party itself. The communication analysis in the initial registration process and in subsequent communications is described. Finally I created laboratory exercises with a focus on the Open IMS Core System, where students learn about architecture and principle of networks based on IMS technology operation, with individual elements necessary for the operation of the network and their configuration. Students also test simple captured traffic analysis.

Keywords: IMS network, SIP protocol, security, registration, subscriber, signalization

PORUBSKÝ, T. Řešení bezpečnosti v IMS. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 119s. Vedoucí diplomové práce Ing. Tomáš Mácha.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma "Řešení bezpečnosti v IMS" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Tomáši Máchovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování mé diplomové práce.

V Brně dne ....

podpis autora

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

V dnešní době již poskytovatelé telekomunikačních služeb implementovali do svých vybudovaných sítí většinu služeb, jenž ve svých sítích mohou provozovat s ohledem na přiměřené náklady. Nové služby, které by rádi poskytovali svým zákazníkům, však vyžadují jiný způsob přenosu a zaměřují se zejména na využití IP protokolu (Internet Protocol). Implementace takových služeb však představuje nemalé investice do infrastruktury sítě poskytovatele telekomunikačních služeb. Řešením, které by co nejefektivněji implementovalo tyto služby, by mohl být systém zvaný IMS (IP Multimedia Subsystem).

IMS je architektura, jenž má nahradit současné rozdělení na sítě s přepínáním okruhů (CS - Circuit Switched) a sítě s přepínáním paketů (PS - Packet Switched). Hlasové služby jsou dnes většinou realizovány přepínáním okruhů v CS doméně, kdežto přenos dat je realizován přepínáním paketů v PS doméně.

IMS sjednocuje přenos hlasu i dat do paketů, čímž zachovává současné schéma pro přenos dat a přenos hlasu převádí na VoIP platformu protokolu SIP, kde zavádí kvalitu řízení přenosu a priorizaci VoIP, tedy obdobu QoS (Quality of Service) v IP sítích.

Cílem mé diplomové práce je osvětlení technologie IMS, kde se z počátku zaměřuji na architekturu tohoto systému. Popisuji zde hlavní prvky systému IMS, kterými jsou *HSS* - hlavní databáze informací o uživatelích, *SLF* - jednoduchá databáze mapující adresy uživatelů do HSS, dále pak *CSCF* - SIP server, zpracovávající SIP signalizaci v IMS, *AS* - aplikační server hostující a vykonávající služby, *MRF* - poskytující zdroj prostředků v domácí síti, *BGCF* - jakožto SIP server obsahující směrovací funkce založené na telefonních číslech, *IMS-ALG* a *TrGW* - entity služeb provádějící překlad mezi protokoly IPv4 a IPv6 a *PSTN/CS* - poskytující rozhraní do sítě s přepínáním okruhů.

Dále popisuji registraci v systému IMS s výpisem přenášených zpráv nutných pro tuto registraci. V této kapitole také vypisuji všechna rozhraní, které se v systému IMS nachází spolu s použitými protokoly, kterých je zde opravdu mnoho. Poté se zaměřuji na problematiku bezpečnosti v IMS. Bezpečnost zde můžeme rozdělit na *zabezpečení přístupu*, kde se využívá autentizace (ověření identity uživatele služeb) a autorizace (přidělení oprávnění přístupu k určitým službám), a na *síťovou bezpečnost*, která se zabývá zabezpečením provozu mezi odlišnými zabezpečenými doménami.

Následující kapitola je věnována systému Open IMS Core, což je open source projekt Fraunhoferova Institutu FOKUS (Fraunhofer Institut für Offene Kommunikationssysteme), jenž poskytuje základní implementaci pro testování IMS technologie. Je zde popsána obsluha tohoto systému od její instalace, přes konfiguraci jednotlivých serverů, všech potřebných nastavení a dodatečných instalací, až k samotné komunikaci mezi jednotlivými uživateli v nakonfigurované IMS síti.

Poté se blíže věnuji analýze přenášených zpráv v této síti, kde podrobně popisuji počáteční registraci účastníka IMS sítě a dále sestavení, průběh a ukončení spojení mezi jednotlivými účastníky.

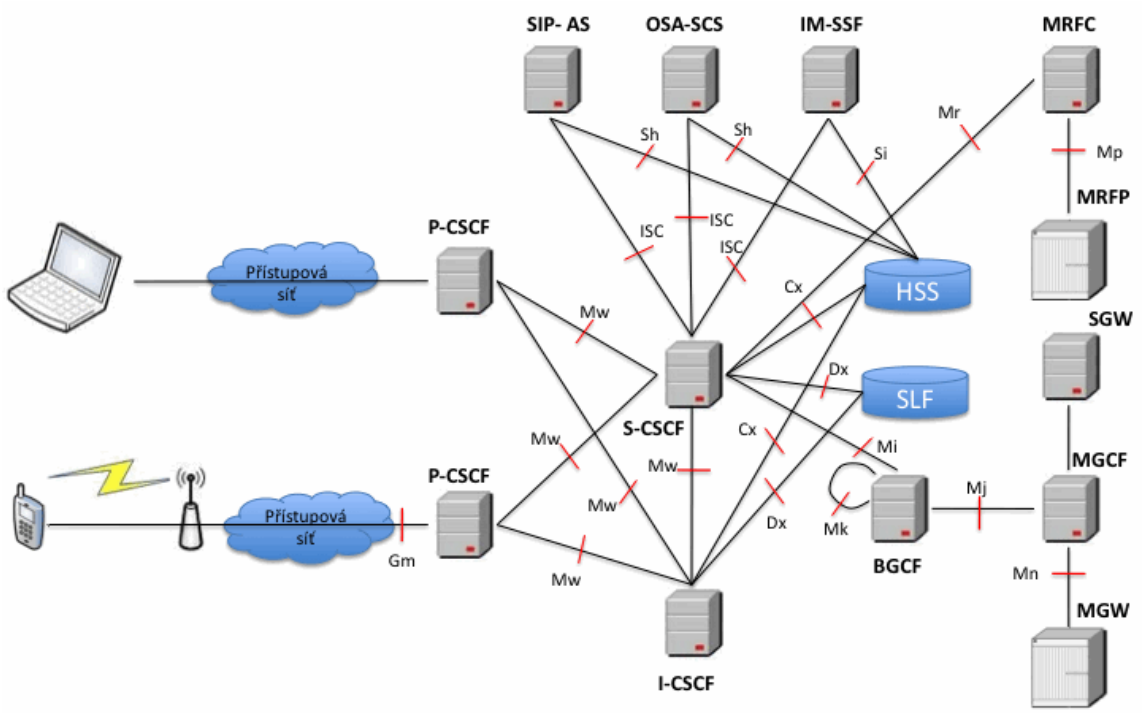
V závěru mé diplomové práce navrhuji laboratorní úlohu, zaměřenou na práci v open source systému Open IMS Core. Cílem této úlohy je seznámit studenty s architekturou a principem fungování sítí založených na technologii IMS, praktické seznámení s jednotlivými prvky, nutnými pro provoz této sítě a jejich základním nastavením. Studenti si dále vyzkouší jednoduchou analýzu zachycené komunikace.

2 Architektura IMS

Než popíši obecnou architekturu IMS, tak je důležité zmínit, že projekt 3GPP (The 3rd Generation Partnership Project) nestandardizuje uzly, ale služby. To tedy znamená, že IMS je souhrn služeb spojených pomocí standardizovaných rozhraní. Vývojáři tak mohou spojit dvě služby do jednoho uzlu. Podobně mohou také rozdělit jednu službu mezi dva a více uzly.

Obecně lze říci, že většina výrobců sleduje IMS architekturu podrobně a každou službu implementují do samostatného uzlu. Nicméně je možné najít i uzly, které implementují více než jednu službu a služby distribuované přes více než jeden uzel.

[2], [3], [5], [7]



Obr. 2-1 Architektura IMS

Na Obr. 2-1 je vidět architektura IMS jak ji standardizuje 3GPP. Je zde vyobrazena většina signalizačních rozhraní z IMS (Cx, Dx, Sh atd.). Nejsou tu všechny, ale jen ty nejdůležitější.

Na levé straně Obr. 2-1 je mobilní terminál, typicky se mu říká User Equipment (UE). Ten se připojuje do paketové sítě, jakou je GPRS, přes rádiové spojení. Nicméně IMS podporuje i jiné typy zařízení a přístupů. Např. PDA (Personal Digital Assistant) a počítače se mohou také připojit do IMS sítě. Příkladem dalších přístupů jsou WLAN a ADSL.

Uzly v síti zvané IP Multimedia Core Network Subsystem jsou následující:

- jedna nebo více databází uživatelů zvaná HSS (Home Subscriber Server) a SLF (Subscriber Location Functions) - jednoduchá databáze mapující adresy uživatelů do HSS,
- jeden nebo více SIP serverů, souhrnně označováno CSCF (Call/Session Control Functions),
- jeden nebo více aplikačních serverů AS (Application Servers),
- jeden nebo více MRF (Media Resource Functions) - prostředky pro multimediální služby, ty se pak dále dělí na MRFC (Media Resource Functions Controllers) a MRFP (Media Resource Functions Processors).
- jeden nebo více BGCF (Breakout Gateway Control Functions) - SIP server odpovědný za výběr bodu přestupu do CS domény nebo jiného BGCF,
- jedna nebo více PSTN brán (gateways), každá rozdělena na SGW (Signaling Gateway), MGCF (Media Gateway Control Function) a MGW (Media Gateway).

Ještě je třeba zmínit, že architektura na obrázku nezahrnuje rozhraní spojené s účtováním.

2.1 Databáze: HSS a SLF

Home Subscriber Server (HSS) je centrální úložiště pro informace týkající se uživatelů. Technicky se jedná o odvození z HLR (Home Location Server), který představuje uzel v síti GSM. HSS obsahuje všechna předplacená uživatelská data pro zpracování multimediálního spojení. Tyto data zahrnují kromě jiného informace o poloze, informace o zabezpečení (autorizace a autentizace), informace o profilu uživatele (obsahující předplacené služby uživatele) a S-CSCF (Serving-CSCF) přidělený uživateli.

Síť může obsahovat i více než jen jeden HSS, a to případech, kdy je počet účastníků příliš velký na to, aby byl řízen jedním HSS. V některých případech jsou všechna data související s jednotlivými uživateli uložena v jediném HSS.

Sítě s jedním HSS nepotřebují Subscription Locator Function (SLF), naopak sítě s více než jedním HSS tento SLF vyžadují. SLF je jednoduchá databáze, která mapuje adresy uživatelů do HSS. Uzel, který pošle dotaz na SLF s adresou uživatele na vstupu, obdrží HSS, jenž obsahuje veškeré informace spojené s daným uživatelem. Jak HSS, tak i SLF implementují protokol Diameter se specifickými IMS Diameter aplikacemi.

[2]

2.2 Call Session Control Function

CSCF (Call/Session Control Function), jakožto SIP server, je nezbytný uzel v IMS. Zpracovává SIP signalizaci v IMS a jsou zde tři typy CSCF podle toho, jakou funkci provozují. Všechny se souhrnně označují jako CSCF a dále se dělí do těchto tří kategorií:

- P-CSCF (Proxy-CSCF),
- I-CSCF (Interrogating-CSCF),
- S-CSCF (Serving-CSCF).

[2], [3], [8], [11]

2.2.1 Proxy-CSCF

P-CSCF je z pohledu signalizace prvním bodem kontaktu mezi IMS terminálem a IMS sítí. Z pohledu SIP vystupuje P-CSCF jako příchozí/odchozí SIP proxy server. To znamená, že všechny žádosti od IMS terminálu nebo žádosti směřující do IMS terminálu prochází přes P-CSCF. P-CSCF pak tyto SIP žádosti/odpovědi zasílá příslušnou cestou, a to směrem k IMS terminálu nebo k IMS síti.

P-CSCF je přidělen IMS terminálu během IMS registrace a po celou dobu registrace se nemění, takže IMS terminál po celou dobu registrace komunikuje jen s jedním P-CSCF.

P-CSCF má několik funkcí, z nichž některé jsou spojené se zabezpečením. Nejdříve stanoví číslo IPsec zabezpečeného spojení směrem k IMS terminálu. Toto IPsec spojení nabízí zajištění integrity, a tak je schopné rozpoznat jakékoliv změny ve zprávě od jejího vzniku.

Jakmile P-CSCF autentizuje uživatele, což je součást založení bezpečného spojení, tak pošle tuto identitu uživatele ostatním uzlům v síti. Tyto uzly už pak nemusí znovu autentizovat tohoto uživatele, protože důvěřují P-CSCF. Zbytek uzlů v síti má řadu účelů, jako třeba poskytování osobních služeb nebo vytváření záznamů účtů. P-CSCF pak dodatečně ověřuje správnost SIP žádostí zaslaných IMS terminálem. Pomocí tohoto ověření zabrání IMS terminálu vytvářet SIP žádosti, které nejsou v souladu s pravidly SIP.

P-CSCF umožňuje také kompresi a dekompresi SIP zpráv. Tyto zprávy mohou být objemné, protože SIP je textově orientovaný protokol. Zatímco mohou být SIP zprávy přenášeny přes širokopásmové spojení v dosti krátkém čase, tak přenos objemných SIP zpráv přes úzkopásmové spojení, jako jsou některé rádiové spoje, může trvat i pár sekund. K redukci takového zpoždění využijeme metody komprese a zašleme tak tyto zprávy do cílového bodu, kde dochází k dekompresi.

P-CSCF může také obsahovat službu pro rozhodování o politice - PDF (Policy Decision Function), a to jako součást P-CSCF nebo jako samostatnou jednotku. PDF autorizuje rovinu zdrojů prostředků a řídí QoS (Quality of Service), tedy kvalitu služeb.

P-CSCF také generuje informace o účtování pro uzel, ve kterém se tyto informace shromažďují.

IMS obvykle obsahuje více P-CSCF kvůli rozšiřitelnosti a zálohování. Jednotlivé P-CSCF obsluhují několik IMS terminálů, to závisí na kapacitě uzlů.

Umístění P-CSCF

P-CSCF se může nacházet buď v navštívené síti nebo v domácí síti. V případě výchozí paketové sítě založené na GPRS je často umístěn ve stejné síti spolu s GGSN (Gateway GPRS Support Node). Jak P-CSCF tak GGSN jsou tedy umístěny buď v navštívené síti nebo v domácí síti.

2.2.2 Interrogating -CSCF

I-CSCF je SIP proxy server umístěný na okraji administrativní domény. Adresa I-CSCF je uvedena v DNS (Domain Name System) záznamech. Kromě funkcí SIP proxy serveru má I-CSCF také rozhraní pro přístup k SLF a HSS. Tyto rozhraní jsou založeny na protokolu Diameter. I-CSCF obdrží informace o poloze uživatele a směruje SIP žádosti příslušnou cestou (typicky do S-CSCF).

I-CSCF také může volitelně šifrovat části SIP zpráv, které obsahují citlivé informace o doméně, jako např. počet serverů v doméně, jejich DNS jména nebo jejich kapacity. Tato funkce se označuje jako THIG (Topology Hiding Inter-network Gateway), je volitelná a pravděpodobně nebude nabízena mnoha sítěmi.

Síť bude většinou obsahovat více I-CSCF kvůli rozšiřitelnosti a zálohování.

Umístění I-CSCF

I-CSCF se většinou vyskytuje v domácí síti, i když v některých výjimečných případech, jako např. I-CSCF (THIG), se může vyskytovat v navštívené síti.

2.2.3 Serving -CSCF

Jedná se o centrální bod na signalizační úrovni. Je to vlastně SIP server, ale provádí také řízení spojení. Kromě funkcí SIP serveru funguje S-CSFC také jako SIP registrátor. To znamená, že si udržuje spojení mezi lokací uživatele, např. IP adresu terminálu, ke kterému je uživatel přihlášen, a mezi uživatelovou adresou SIP záznamu - známou také jako veřejná identita uživatele - Public User Identity. Stejně jako u I-CSCF je i zde použito směrem k HSS rozhraní Diameter.

Hlavní účely tohoto rozhraní jsou následující:

- stáhnutí autentizačních vektorů z HSS pro uživatele, který se snaží vstoupit do IMS,
- stáhnutí profilu uživatele z HSS. Tento profil zahrnuje profil služeb, což je sada spouštěčů, které mohou způsobit, že se budou SIP zprávy směrovat přes jeden nebo více aplikačních serverů,
- informování HSS o přiděleném S-CSCF uživateli po dobu této registrace.

Všechny SIP zprávy signalizující, že IMS terminál odesílá nebo přijímá, projdou přes S-CSCF. S-CSCF zkontroluje každou SIP zprávu a rozhodne, zda by měla SIP signalizace projít přes jeden nebo více aplikačních serverů na cestě do konečného místa určení. Tyto aplikační servery by potenciálně měly poskytovat služby uživatelům.

Jedna z hlavních funkcí S-CSCF je poskytovat služby směrování SIP zpráv. Pokud uživatel vytočí telefonní číslo namísto SIP URI (Uniform Resource Identifier), tak S-CSCF provede překlad, typicky založený na standardu DNS E.164 Number Translation.

S-CSCF také uplatňuje politiku operátora sítě, např. že uživatel nemusí být pro určité typy spojení autorizován.

V síti je většinou několik S-CSCF kvůli rozšiřitelnosti a zálohování. Každý S-CSCF obsluhuje několik IMS terminálů, což závisí na kapacitě uzlů.

Umístění S-CSCF

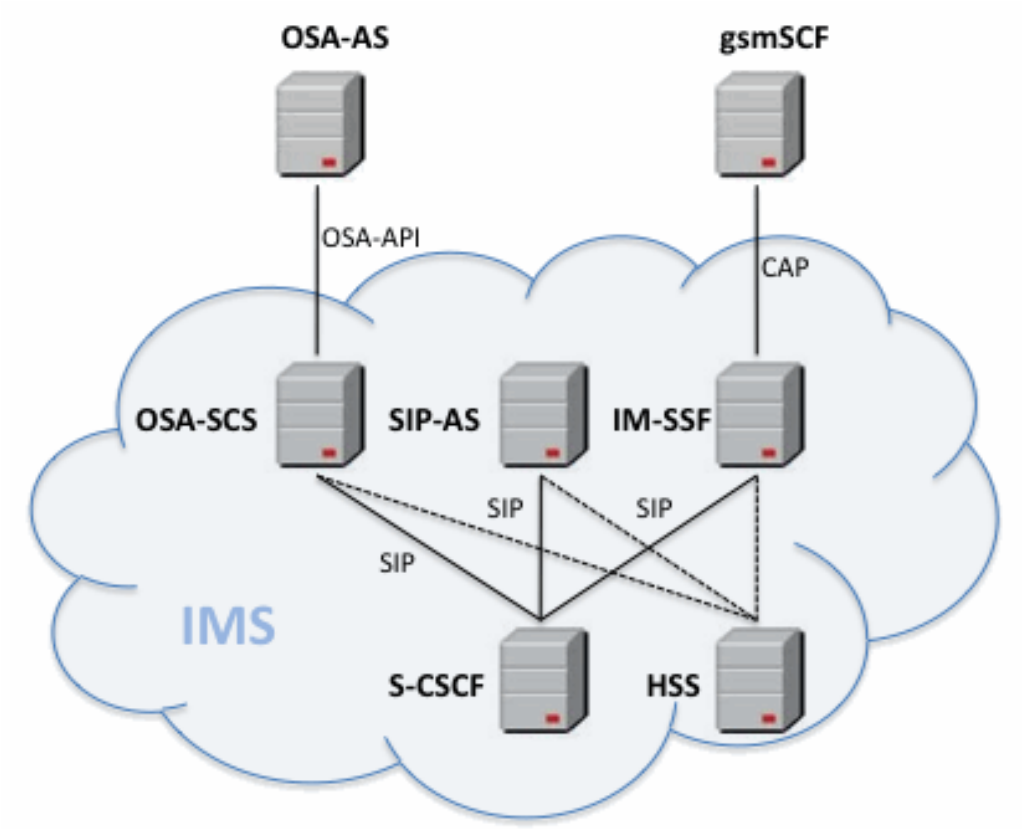
S-CSCF se vždy nachází v domácí síti.

2.3 Application Server

AS (Application Server) je SIP objekt, který hostuje a vykonává služby. V závislosti na aktuální službě může AS pracovat v následujících módech:
- SIP proxy módu,
- SIP UA (User Agent) módu - např. koncový bod,
- SIP B2BUA (Back-to-Back User Agent) módu - např. vzájemné spojení dvou SIP UA.

AS se připojuje k S-CSCF pomocí protokolu SIP. Na Obr. 2-2 můžeme vidět tři typy těchto aplikačních serverů.

[2], [3], [4]



Obr. 2-2 Tři typy aplikačních serverů

SIP AS (Application Server)

Jedná se o nativní aplikační server, který hostuje a vykonává IP multimediální služby založené na SIP.

OSA-SCS (Open Service Access-Service Capability Server)

Tento aplikační server poskytuje rozhraní do soustavy OSA aplikačních serverů. Zdědil veškeré možnosti OSA, zvláště možnost bezpečného přístupu do IMS z externích sítí. Tento uzel představuje na jedné straně aplikační server (spojující S-CSCF pomocí protokolu SIP) a na druhé straně rozhraní mezi OSA aplikačním serverem a OSA programovacím aplikačním rozhraním - OSA API (Application Programming interface).

IM-SFF (IP Multimedia Service Switching Function)

Tento speciální aplikační server nám dovoluje použít CAMEL (Customized Applications for Mobile network Enhanced) služby - uživatelské aplikace pro rozšířené mobilní sítě, které byly vyvinuty pro GSM v IMS. IM-SFF poskytuje gsmSCF (GSM Service Control Function) - funkce řízení služeb, pro řízení IMS spojení. IM-SFF představuje na jedné straně aplikační server (spojující S-CSCF pomocí protokolu SIP) a na druhé straně SSF (Service Switching Function) - funkce přepínání služeb, spojující gsmSCF protokolem založeném na CAP (CAMEL Application Part).

Všechny tyto tři typy aplikačních serverů představují pro IMS síť SIP aplikační servery (tzn. buď SIP proxy server, SIP User Agent, SIP redirect server nebo SIP Back-to-Back User Agent).

AS může volitelně poskytovat rozhraní k HSS. SIP-AS a OSA-SCS rozhraní k HSS jsou založeny na protokolu Diameter a slouží k stáhnutí nebo vysílání dat, která jsou spojená s uživatelem a uložená v HSS. IM-SFF rozhraní k HSS je založeno na protokolu MAP (Mobile Application Part).

Umístění AS

AS se může nacházet buď v domácí síti nebo v externí síti třetí strany, se kterou má domovský operátor smlouvu o službách. V každém případě, pokud se AS nachází mimo domovskou síť, tak se k HSS nepřipojuje.

2.4 Media Resource Function

MRF (Media Resource Function) poskytuje zdroj prostředků v domácí síti. Poskytují míchání proudu médií, převod mezi různými kodeky, získání statistik a provádí všechny druhy analýz těchto prostředků.

MRF se dále dělí na signalizační rovinu uzlu zvanou MRFC (Media Resource Function Controller) a na rovinu prostředků uzlu zvanou MRFP (Media Resource Function Processor). MRFC působí jako SIP User Agent a obsahuje SIP rozhraní k S-CSCF. MRFC řídí prostředky v MRFP přes rozhraní H.248. MRFP provádí veškeré služby spojené s médii, jako je přehrávání a míchání médií.

[2], [3], [8]

Umístění MRF

MRF se vždy nachází v domovské síti.

2.5 Breakout Gateway Control Function

BGCF (Breakout Gateway Control Function) je v podstatě SIP server, který obsahuje směrovací funkce založené na telefonních číslech. BGCF se používá pouze u spojení inicializovaného IMS terminálem a adresované uživateli do sítě s přepínáním okruhů, jako je PSTN nebo PLMN. Hlavní funkce BGCF jsou:

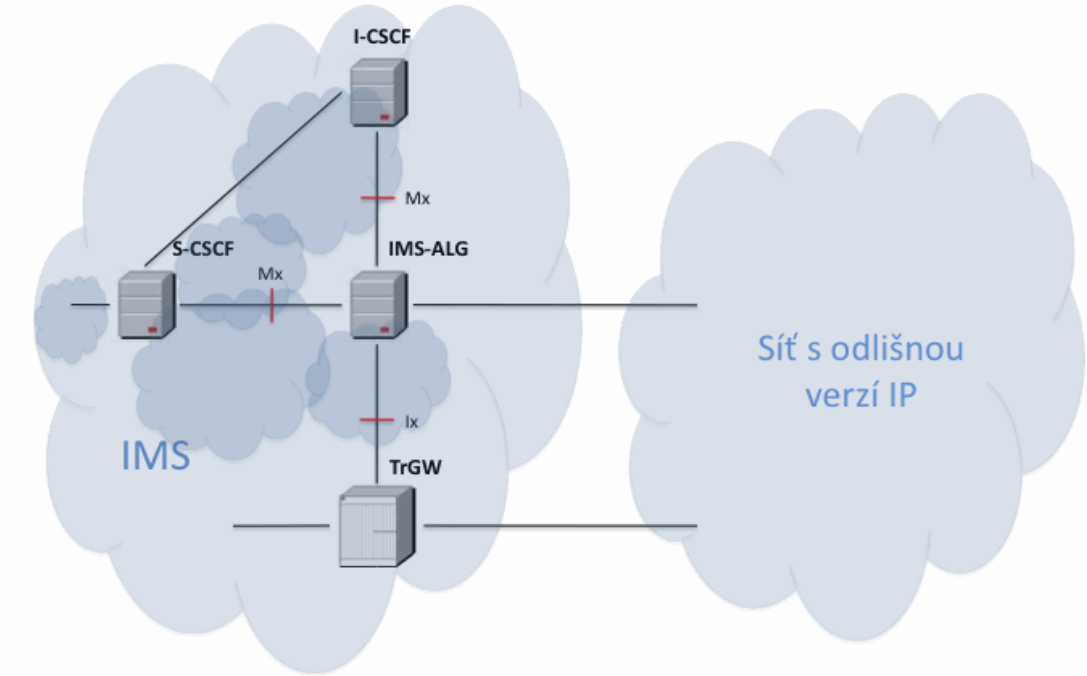
- výběr vhodné sítě, kde se vyskytuje vzájemná komunikace domén s přepínáním okruhů,
- výběr vhodné PSTN/CS brány (gateway), pokud probíhá vzájemná komunikace ve stejné síti, ve které se nachází BGCF.

[2]

2.6 IMS-ALG a TrGW

IMS podporuje dvě verze IP protokolu - IPv4 a IPv6. V některých částech IP multimediálního spojení se může vyskytnout vzájemná komunikace mezi těmito dvěma protokoly. Ve snaze usnadnit vzájemnou komunikaci mezi IPv4 a IPv6, aniž by bylo zapotřebí vyžadovat podporu u terminálů, přidává IMS dvě nové entity služeb, které provádí překlad mezi oběma protokoly. Těmito entitami jsou IMS Application Layer Gateway (IMS-ALG) - brána aplikační vrstvy IMS a Transition Gateway (TrGW) - brána přechodu. Původně zpracovávala signalizaci na řídící úrovni - např. SIP a SDP zprávy, později zpracovávala přenos na uživatelské úrovni - např. RTP, RTCP.

[2], [3]



Obr. 2-3 IMS-ALG a TrGW

Obr. 2-3 ukazuje spojení IMS-ALG s TrGW a s ostatními IMS uzly. Brána IMS-ALG zde působí jako SIP B2BUA udržováním dvou nezávislých větví: jednou do vnitřní IMS sítě a druhou do další sítě. Každá tato větev běží na jiné verzi IP protokolu. Brána IMS-ALG pak dodatečně přepíše SDP (Session Description Protocol) záměnou IP adres a čísel portů vytvořených terminálem za jednu nebo více IP adres a čísla portů přidělené bráně TrGW. To umožní směrovat přenos na uživatelské úrovni do brány TrGW.

Brána IMS-ALG je pro příchozí provoz spojen se serverem I-CSCF a pro odchozí provoz se serverem S-CSCF přes rozhraní Mx.

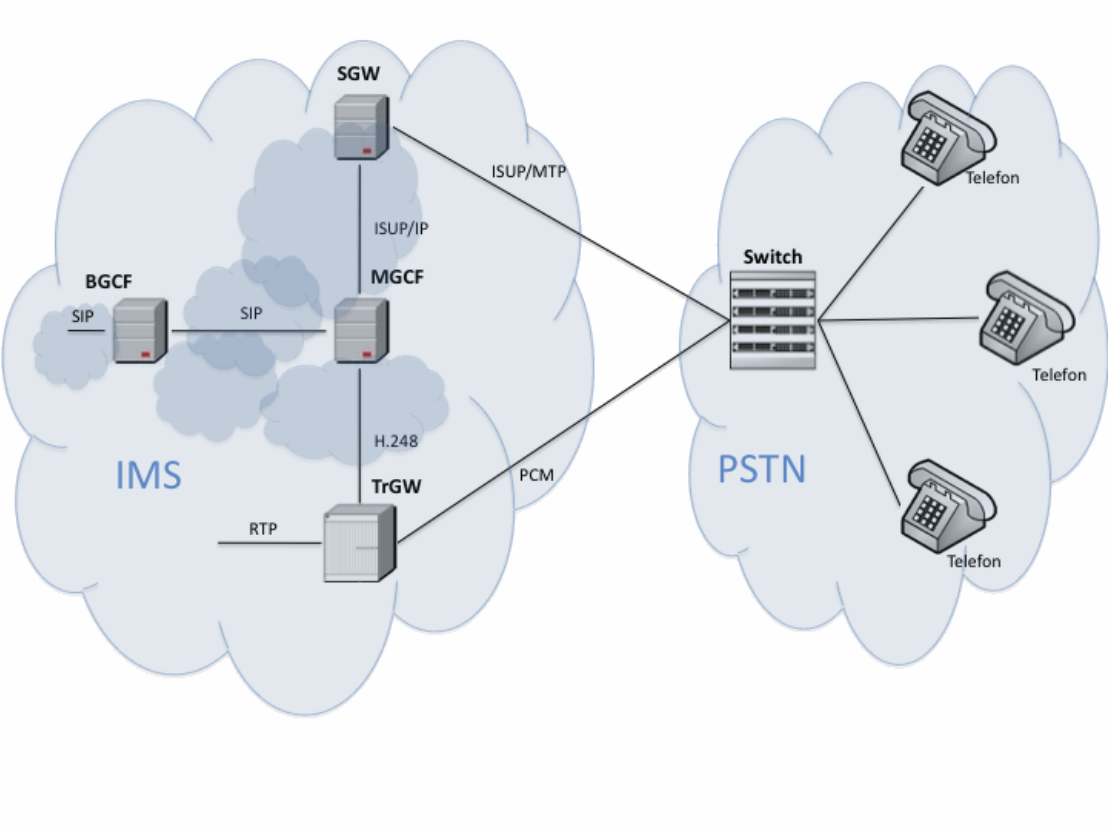
Brána TrGW je účinný NAT-PT/NAPT-PT (Network Address Port Translator - Protocol Translator), tedy překladač adres a portů a konfiguruje se společnými IPv4 adresami, které jsou dynamicky přidělované pro danou relaci. TrGW překládá IPv4 a IPv6 na mediální úroveň (např. RTP, RTCP).

Brána IMS-ALG je spojena s bránou TrGW přes rozhraní Ix.

2.7 PSTN/CS Gateway

Brána PSTN (Public Switched Telephone Network) poskytuje rozhraní do sítě s přepínáním okruhů a dovoluje IMS terminálům vytvořit/přijmout hovor do/z PSTN, nebo některé z ostatních sítí s přepínáním okruhů. Na Obr. 2-4 vidíme BGCF a PSTN bránu, která spojuje PSTN.

[2], [3], [9]



Obr. 2-4 PSTN/CS brána spojující síť s přepínáním okruhů

PSTN brána je rozdělena do následujících funkcí:

SGW (Signaling Gateway)

Spojuje rovinu signalizace CS (Circuit Switched) sítě (např. PSTN). SGW provádí konverzi protokolu na spodní vrstvě. Nahrazuje nižší MTP (Message Transfer Part) přenos za SCTP (Stream Control Transmission Protocol) přes IP. To znamená, že SGW provádí konverzi přenosu ISUP nebo BICC přes MTP za ISUP nebo BICC přenos přes SCTP/IP.

MGCF (Media gateway Control Function)

Jedná se o centrální uzel PSTN/CS brány. Implementuje prostředky pro konverzi protokolů a mapování protokolu SIP (řídícího protokolu hovoru na straně IMS) do ISUP přes IP nebo do BICC přes IP (BICC i ISUP jsou řídící protokoly hovorů v sítích s přepínáním okruhů). Kromě konverze těchto protokolů také řídí prostředky MGW (Media Gateway). Použitý protokol mezi MGCF a MGW je H.248.

MGW (Media gateway)

Spojuje mediální úroveň (rovinu prostředků) PSTN nebo CS sítě. Na jedné straně je MGW schopno poslat/přijmout IMS prostředky přes Real-Time Protocol (RTP), na druhé straně MGW používá jeden nebo více časových slotů pulzní kódové modulace PCM (Pulse Code Modulation) pro spojení s CS sítí. V případě že IMS terminál nepodporuje kodek použitý CS stranou, tak ho převede. To nastane, když IMS terminál používá AMR kodek a PSTN terminál používá kodek G.711.

3 Systém IMS

Tato kapitola popisuje registraci a ustálení spojení v IP Multimedia Subsystem (IMS).

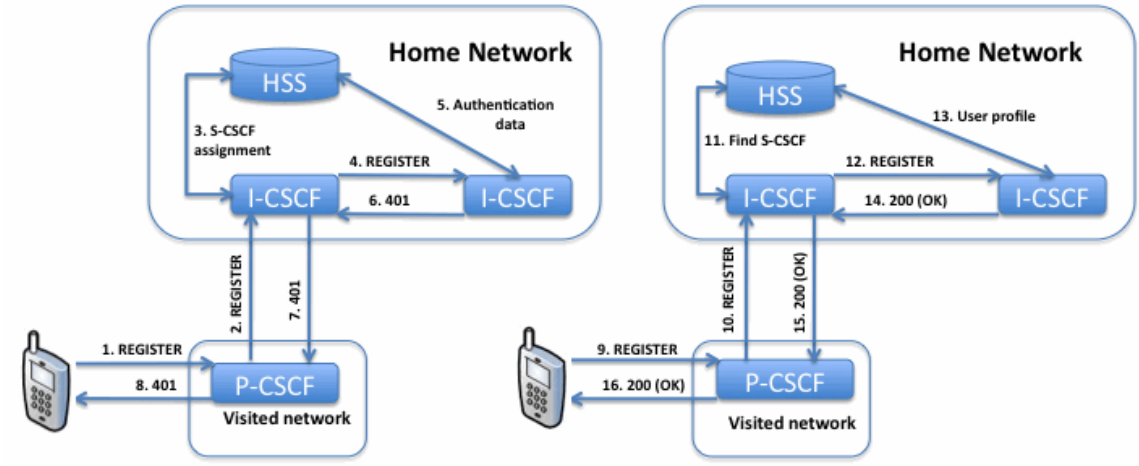
Ještě než se provede IMS registrace, musí User Equipment (UE) nalézt IMS objekty, kterým pošle žádost *REGISTER*. Tento systém se jmenuje nalezení Proxy-Call Session Control Function (P-CSCF). Dále potřebuje UE získat identitu uživatele z identifikační jednotky. Během registrace bude přidělen Serving-CSCF (S-CSCF), provede se autentizace a bude ustáleno příslušné zabezpečené spojení, stáhne se profil uživatele do přiděleného S-CSCF, inicializuje se Session Initiation Protocol (SIP) komprese a budou dodány implicitně registrované veřejné identity uživatelů.

[1]

3.1 Registrace

Ještě před IMS registrací, která dovolí UE využívat IMS služby, musí UE získat IP spojení a nalézt vstupní bod do IMS, tedy P-CSCF. Například, u přístupu přes General Packet Radio Service (GPRS), vykonává UE proceduru GPRS připojení a aktivuje protokol Packet Data Protocol (PDP) pro SIP signalizaci.

[1], [3]



Obr. 3-1 IMS registrace

IMS registrace má dvě fáze: na levé straně Obr. 3-1 je vidět, jak síť vyzývá UE, pravá strana zase ukazuje odpověď UE na tuto výzvu a dokončení registrace.

UE nejprve pošle SIP žádost *REGISTER* nalezenému P-CSCF. Tato žádost by měla obsahovat, pod jakou identitou se bude registrovat a adresu domácí domény (adresu dotazovaného CSCF nebo I-CSCF). P-CSCF zpracovává žádosti *REGISTER* a používá poskytnutou domovskou doménu k získání IP adresy I-CSCF. I-CSCF pak kontaktuje Home Subscriber Server (HSS) k získání požadovaných schopností pro výběr S-CSCF. Po výběru S-CSCF pošle I-CSCF žádost *REGISTER* do S-CSCF. Ten zjistí, že uživatel není registrovaný, a tak si vezme autentizační data z HSS a pošle uživateli odpověď *401* neautorizován. Podruhé už bude uživatel počítat s odpovědí na výzvu a pošle další žádost *REGISTER* do P-CSCF. P-CSCF znovu nalezne I-CSCF a ten pak nalezne S-CSCF.

Nakonec P-CSCF zkontroluje odpovědi a pokud je vše v pořádku, stáhne profil uživatele z HSS a potvrdí registraci odpovědí *200 OK*. Pokud je UE úspěšně autorizován, může začít přijímat spojení. UE i P-CSCF se během registrační procedury dozví, které S-CSCF v síti bude sloužit pro UE. Odpovědností UE je, aby zachoval svou registraci aktivní tím, že ji bude periodicky obnovovat. Pokud by tak neučinil, tak by S-CSCF po vypršení časového intervalu jeho registraci bez varování smazal. Pokud chce UE zrušit registraci u IMS, stačí nastavit časovou prodlevu na 0 a poslat žádost *REGISTER*.

Informace uložené v paměti před, během a po registraci:

UE

Před registrací:	adresa P-CSCF, domovská doména, kredit, veřejná a soukromá identita uživatele.
Během registrace:	adresa P-CSCF, domovská doména, kredit, veřejná (a implicitně registrované veřejné identity) a soukromá identita uživatele, zabezpečené spojení.
Po registraci:	adresa P-CSCF, domovská doména, kredit, veřejná a soukromá identita uživatele, zabezpečené spojení, směrovací informace (S-CSCF).

P-CSCF

Před registrací:	žádné .
Během registrace:	výchozí vstupní bod sítě, IP adresa UE, veřejné a soukromé ID uživatele, zabezpečené spojení.
Po registraci:	konečný vstupní bod sítě S-CSCF, adresa UE, registrovaná veřejná identita uživatele (a implicitně registrované veřejné identity uživatelů), soukromé ID uživatele, zabezpečené spojení, adresa služeb účtování za data CDF (Charging Data Function).

I-CSCF

Před registrací:	adresa HSS nebo SLF .
Během registrace:	HSS nebo SLF záznam, adresa P-CSCF a S-CSCF.
Po registraci:	adresa HSS a SLF.

S-CSCF

Před registrací:	adresa HSS nebo SLF.
Během registrace:	adresa a jméno HSS, profil uživatele, adresa a jméno proxy, veřejné a soukromé ID, IP adresa UE.
Po registraci:	adresa a jméno HSS, profil uživatele, adresa a jméno proxy, veřejné a soukromé ID, IP adresa UE.

HSS

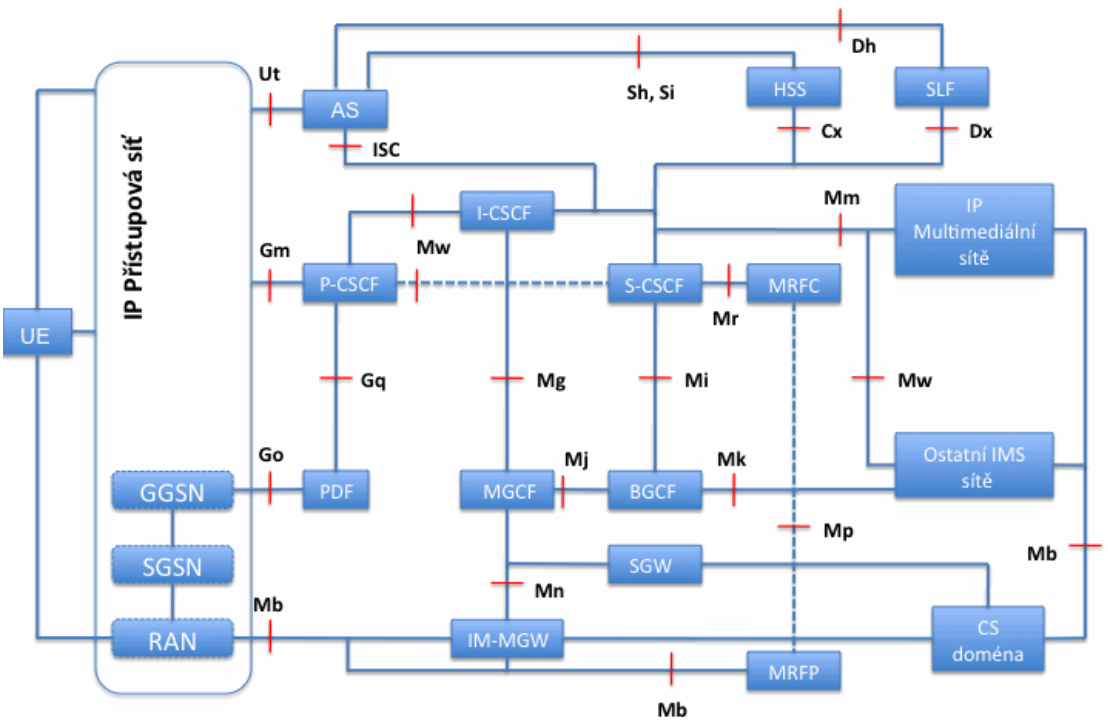
Před registrací:	profil uživatele, zvolené parametry S-CSCF.
Během registrace:	profil uživatele, zvolené parametry S-CSCF, informace o navštívených sítích, pokud se uživatel pohybuje.
Po registraci:	profil uživatele, zvolené parametry S-CSCF, informace o tom, které identity uživatele jsou registrovány, jméno S-CSCF přidělené uživateli.

3.2 Typy rozhraní v IMS

Zde jsou popsána jednotlivá propojení různých částí sítí a použité protokoly. Architekturu IMS můžeme vidět na Obr. 3-2. Ta je z důvodu srozumitelnosti zjednodušena:

- nezahrnuje služby spojené s účtováním a jejich rozhraní,
- nezahrnuje různé druhy AS (application server),
- nezahrnuje spojení mezi různými IMS sítěmi a AS,
- nezahrnuje SEG (Security Gateway) na rozhraních Mm, Mk, Mw,
- tečkovaná čára mezi objekty značí přímé spojení,
- IMS Service Control (ISC), cx, Dx, Mm a Mw ukončuje jak S-CSCF tak i I-CSCF.

[1], [3], [9]



Obr. 3-2 Architektura sítě IMS včetně rozhraní

Rozhraní Gm

Rozhraní Gm spojuje UE do IMS. Používá se k přenosu všech SIP signalizačních zpráv mezi UE a IMS. Na straně IMS vede toto rozhraní do P-CSCF. Procedury na rozhraní Gm mohou být rozděleny do tří kategorií: registrace, řízení spojení a přenos.

Při registraci používá UE rozhraní Gm pro zaslání žádosti o registraci, spolu s údaji o podporovaných bezpečnostních mechanismech, do P-CSCF. Během registrace si UE vzájemně se sítí vymění nezbytné parametry k autentizaci, získá implicitně registrovanou identitu uživatele, s P-CSCF dohodne nezbytné parametry pro zabezpečené spojení - SA (Security Association) a vhodnou počáteční SIP kompresi. Navíc se ještě toto rozhraní používá k informování UE o jeho odhlášení vyvolané sítí nebo při obnově autentizace vyvolané sítí.

Procedura řízení spojení obsahuje mechanismy jak pro vytvoření spojení mobilním telefonem, tak pro ukončení spojení mobilním telefonem. V prvním případě slouží Gm rozhraní k předání žádosti z UE do P-CSCF, v druhém případě naopak z P-CSCF do UE.

Procedury přenosu slouží k posílání samostatných žádostí (např. *MESSAGE*) a pro přijetí všech odpovědí (např. *200 OK*) na tyto žádosti přes rozhraní Gm. Rozdíl mezi procedurou řízení spojení a procedurou přenosu je v tom, že se nevytváří dialog.

Rozhraní Mw

Rozhraní Gm spojuje UE do IMS, konkrétně do P-CSCF. Dále je zapotřebí rozhraní založené na protokolu SIP pro spojení různých CSCF. Toto rozhraní se označuje jako Mw. Procedury na tomto rozhraní můžeme rozdělit do tří kategorií: registrace, řízení spojení a přenos.

Při registraci používá P-CSCF rozhraní Mw pro zaslání žádosti o registraci z UE do I-CSCF. I-CSCF pak přes toto rozhraní předá žádost k S-CSCF. Odpověď z S-CSCF nakonec prochází zpátky přes Mw rozhraní. S-CSCF využívá toto rozhraní ještě k informování UE o odhlášení vyvolané sítí, obnově autentizace vyvolané sítí a k informování P-CSCF, že může uvolnit prostředky týkající se jednotlivých uživatelů.

Procedura řízení spojení obsahuje mechanismy jak pro vytvoření spojení mobilním telefonem, tak pro ukončení spojení mobilním telefonem. V prvním případě slouží Mw rozhraní k předání žádosti jak z P-CSCF do S-CSCF tak i z S-CSCF do I-CSCF. V druhém případě slouží Mw rozhraní k předání žádosti jak z I-CSCF do S-CSCF tak i z S-CSCF do P-CSCF. Dále se toto rozhraní používá také pro uvolnění realizovaného síťového spojení: např. P-CSCF může zahájit uvolnění spojení s S-CSCF, pokud obdrží oznámení od PDF (Policy Decision Function), že doručitel je ztracen. Navíc jsou přes toto rozhraní zprostředkovány informace spojené s účtováním.

Procedury přenosu slouží k průchodu samostatných žádostí (např. *MESSAGE*) a pro přijetí všech odpovědí (*např. 200 OK*) na tyto žádosti přes rozhraní Mw. Jak už bylo řečeno, rozdíl mezi procedurou řízení spojení a procedurou přenosu je v tom, že se nevytváří dialog.

IMS Service Control (ISC) rozhraní

V IMS architektuře jsou AS (Application Server) entity, které podporují služby jako prezence, posílání zpráv a předávání spojení. Proto zde musí být rozhraní pro odesílání a přijímání SIP zpráv mezi CSCF a AS. Toto rozhraní se jmenuje IMS Service Control (ISC) a používá se zde protokol SIP. ISC procedury mohou být rozděleny do dvou hlavních kategorií: směrování výchozích SIP žádostí do AS a iniciace SIP žádostí od AS.

Jakmile S-CSCF obdrží výchozí SIP žádost, tak ji analyzuje. Podle této analýzy může rozhodnout o směrování žádosti do AS k dalšímu zpracování. AS může tyto žádosti od S-CSCF ukončit, přesměrovat nebo zastoupit. AS také může iniciovat žádost místo uživatele.

Cx rozhraní

Účastnická data a data služeb jsou permanentně ukládány v HSS. Tyto centralizovaná data využívají I-CSCF a S-CSCF pokud se uživatel registruje nebo pokud obdrží spojení. Proto musí být mezi HSS a CSCF rozhraní, jenž se nazývá Cx a používá

protokol Diameter. Procedury jsou zde rozděleny do tří kategorií: správa polohy, zpracování uživatelských dat a autentizace uživatele.

Tab. 3-1 uvádí přehled možných Cx příkazů.

Tab. 3-1 Cx příkazy

Příkaz	Účel	Ozn.	Zdroj	Cíl
User-Authorization-Request/Answer	používá se během SIP registrace mezi I-CSCF a HSS pro získání jména nebo schopností S-CSCF k jeho výběru a také pro získání S-CSCF jména během odhlášení	UAR UAA	I-CSCF HSS	HSS I-CSCF
Server-Assignment-Request/Answer	používá se mezi S-CSCF a HSS k aktualizaci jména S-CSCF pro HSS a pro stáhnutí uživatelských dat do S-CSCF	SAR SAA	S-CSCF HSS	HSS S-CSCF
Location-Info-Request/Answer	používá se mezi I-CSCF a HSS během sestavení relace SIP k získání jména S-CSCF, které bude přiřazeno uživateli nebo k získání schopností S-CSCF pro jeho výběr	LIR LIA	I-CSCF HSS	HSS I-CSCF
Multimedia-Auth-Request/Answer	používá se mezi S-CSCF a HSS k výměně informací pro podporu autentizace mezi koncovým uživatelem a domovskou IMS sítí	MAR MAA	S-CSCF HSS	HSS S-CSCF
Registration-Termination-Request/Answer	používá se mezi S-CSCF a HSS pokud HSS administrativně odhlásí jednu nebo více veřejných uživatelských identit	RTR RTA	HSS S-CSCF	S-CSCF HSS
Push-Profile-Request/Answer	používá se mezi HSS a S-CSCF pokud jsou uživatelská data v HSS změněna a potřebují být aktualizována v S-CSCF	PPR PPA	HSS S-CSCF	S-CSCF HSS

Správa polohy

Tyto procedury se ještě dále dělí do dvou skupin: registrace/odhlášení a výběr polohy.

Pokud I-CSCF obdrží od P-CSCF přes Mw rozhraní žádost SIP *REGISTER*, tak použije dotaz na status registrace uživatele nebo známý příkaz dle standardu: *User-Authorization-Request* (UAR). Po obdržení příkazu UAR zašle HSS příkaz *User-Authorization-Answer* (UAA). Ten obsahuje jméno S-CSCF a/nebo možnosti S-CSCF, které závisí na aktuálním stavu registrace uživatele. Pokud uživatel ještě nemá přiděleno jméno S-CSCF v HSS nebo pokud I-CSCF přímo odpoví S-CSCF možnostmi, pak jsou tyto S-CSCF možnosti vráceny. Jinak je vráceno jméno S-CSCF. V případě, že jsou tyto možnosti vráceny, musí I-CSCF provézt výběr S-CSCF.

Poté co I-CSCF najde S-CSCF, jenž bude přidělen danému uživateli, tak mu předá žádost SIP *REGISTER*. Jakmile S-CSCF tuto žádost obdrží, tak používá ke komunikaci s HSS příkaz *Server-Assignment-Request* (SAR). Tímto příkazem informuje HSS o tom, který S-CSCF bude přidělen uživateli, pokud se hodnota vypršení nerovná nule. Naopak pokud se tato hodnota rovná nule, tak se použije příkaz SAR k oznámení, že už S-CSCF nepatří k tomuto uživateli. Předpokladem pro odeslání příkazu SAR je, že uživatel bude úspěšně u S-CSCF autentizován. Po obdržení příkazu SAR odpoví HSS příkazem *Server-Assignment-Answer* (SAA). Ten obsahuje profil uživatele založený na zadaných hodnotách v žádosti SAR a nepovinně adresu funkcí účtování.

Doposud jsem popsal registraci vyvolanou uživatelem a procedury odhlášení (vyvolané uživatelem nebo pomocí S-CSCF) ovládané přes Cx rozhraní. Je zde však stále potřeba dodatečných operací, které zajistí odhlášení ze strany sítě (např. kvůli ukradenému UE). V těchto případech je to HSS, kdo začíná odhlášení ze strany sítě, a to pomocí příkazu *Registration-Termination-request* (RTR). RTR je potvrzený příkazem *Registration-Termination-Answer* (RTA), který jednoduše ukazuje na výsledek operace.

Už jsem popsal, jak I-CSCF používá příkaz UAR k nalezení S-CSCF po obdržení žádosti SIP *REGISTER*. V souvislosti s tím zde musí být také procedura k nalezení S-CSCF, pokud se metoda SIP liší od REGISTER. Touto požadovanou procedurou je využití příkazu *Location-Info-Request* (LIR). HSS pak odpoví příkazem *Location-Info-Answer* (LIA). Tato odpověď obsahuje jméno S-CSCF, nebo možnosti S-CSCF - tyto jsou vráceny, pokud nemá uživatel přiděleno S-CSCF jméno, jinak se vrací SIP URI (Uniform Resource Identifier) z S-CSCF.

Zpracování uživatelských dat

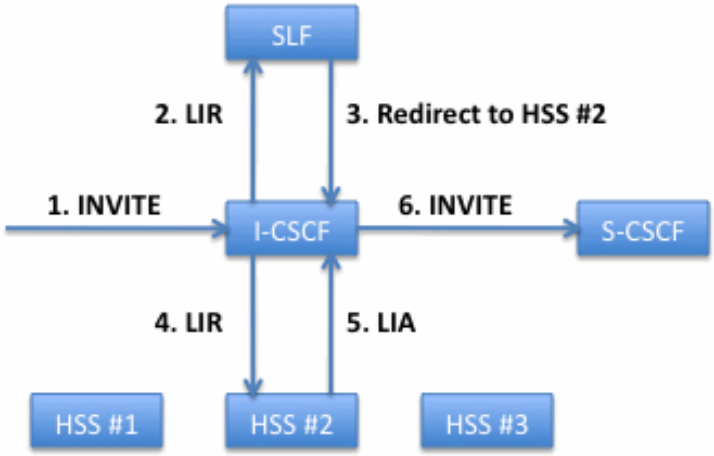
Během registrace se z HSS do S-CSCF stáhnou data související s daným uživatelem a službou, a to přes Cx rozhraní pomocí příkazů SAR a SAA. Tyto data se však můžou časem změnit. Pro aktualizaci dat v S-CSCF zašle HSS příkaz *Push-Profile-Request* (PPR). Aktualizace se provede ihned po změně, avšak pokud S-CSCF obsluhuje nezaregistrovaného uživatele nebo pokud je S-CSCF rezervováno pro nezaregistrovaného uživatele a nastane změna v registrační části profilu uživatele, pak HSS nepošle PPR. PPR příkaz je potvrzen příkazem *Push-Profile-Answer* (PPA), který označuje konec operace.

Autentizační procedury

Autentizace uživatelů je v IMS založena na předem vytvořeném sdíleném klíči. Ten je spolu se sekvencí čísel uložen v IP Multimedia Services Identity Module (ISIM) v UE nebo v HSS této sítě. Protože se S-CSCF stará o autorizaci uživatelů, tak je zde potřeba přenášet přes rozhraní Cx zabezpečená data. Pokud chce S-CSCF autentizovat

uživatele, pošle příkaz *Multimedia-Auth-Request* (MAR) do HSS. HSS pak odpoví pomocí příkazu *Multimedia-Auth-Answer* (MAA). Odpověď obsahuje kromě ostatních informací i autentizační data - autentizační vektor (např. AKAv1 - MD5), autentizační informace (výzva RAND a token AUTN), autorizační informace (očekávaná odpověď nebo XRES), klíč integrity a nepovinně tajný klíč. Dále obsahuje číslo položky, které udává pořadí, ve kterém se budou brát autentizační vektory, pokud jsou vráceny vícenásobné vektory.

Dx rozhraní



Obr. 3-3 Volba HSS pomocí SLF

Pokud jsou v síti rozmístěny vícenásobné a samostatně adresovatelné HSS, pak ani I-CSCF ani S-CSCF neví, který HSS mají kontaktovat. Avšak nejdříve potřebují kontaktovat SLF. Pro tyto účely je zde zavedeno rozhraní Dx. Dx rozhraní se vždy používá ve spojení s rozhraním Cx. Používá se zde protokol Diameter. Jeho funkce je realizována pomocí směrovacího mechanismu prostřednictvím rozšířeného Diameter agenta. K získání adresy HSS zašle I-CSCF nebo S-CSCF do SLF Cx žádosti zaměřené na HSS. Po obdržení adresy HSS z SLF, I-CSCF nebo S-CSCF odešle Cx žádosti do HSS.

Obr. 3-3 ukazuje jak SLF najde správné HSS, pokud I-CSCF obdrží žádost *INVITE* a v síti se nacházejí tři HSS.

Sh rozhraní

AS (SIP AS nebo OSA SCS) může požadovat uživatelská data, nebo potřebuje zjistit kterému S-CSCF zaslat žádost SIP. Tyto informace jsou uloženy v HSS. Proto zde musí existovat rozhraní mezi HSS a AS. Toto rozhraní se označuje jako Sh a používá protokol Diameter. Procedury jsou zde rozděleny do dvou kategorií: zpracování dat a povolení/oznámení. HSS si udržuje seznam AS, které jsou oprávněné k získání nebo ukládání dat. Tab. 3-2 obsahuje přehled možných Sh příkazů.

Tab. 3-2 Sh příkazy

příkaz	účel	zkratka	zdroj	cíl
User-Data-Request/Answer	slouží k poskytování uživatelských dat o konkrétním uživateli	UDR UDA	AS HSS	HSS AS
Profile-Update-Request/Answer	slouží k aktualizaci transparentních dat v HSS	PUR PUA	AS HSS	HSS AS
Subscribe-Notifications-Request/Answer	Subscribe-Notifications Request/Answer příkazy jsou používány k vytvoření/zrušení povolení k uživatelským datům, na kterých jsou vyžadovány oznámení o změnách	SNR SNA	AS HSS	HSS AS
Push-Notification-Request/Answer	slouží k odeslání pozměněných dat do AS	PNA PNR	HSS AS	AS HSS

Zpracování dat

Tyto procedury obsahují možnosti, jak získat uživatelská data z HSS. Tyto uživatelské data můžou obsahovat data týkající se služby (transparentní nebo netransparentní), informace o registraci, identity, jméno S-CSCF které patří k uživateli, adresy služeb účtování a informace o poloze z CS a PS domén. Transparentní data jsou v HSS chápána syntakticky ale ne sémanticky. Naopak netransparentní data jsou v HSS chápána jak syntakticky, tak sémanticky. AS používá jako žádost o data příkaz *User-Data-Request* (UDR). Tato žádost obsahuje informace o požadovaných datech. HSS pak odpoví pomocí *User-Data-Answer* (UDA).

AS může aktualizovat transparentní data v HSS pomocí příkazu *Profile-Update-request* (PUR), který obsahuje data, jenž se mají aktualizovat. Potvrzením pro PUR je pak příkaz *Profile-Update-Answer* (PUA), který značí konec operace.

Povolení /oznámení

Tyto procedury umožní oznámit AS, že jsou určité data pro daného uživatele v HSS aktualizovány. AS zašle *Subscribe-Notification-Request* (SNR), aby obdržel oznámení, jakmile se data v HSS, uvedená v dotazu SNR, změní. HSS pak odpoví pomocí příkazu *Subscribe-Notification-Answer* (SNA), ten značí konec operace.

Pokud AS zaslal příkaz SNR a požadoval oznámení s odsouhlasenou žádostí, pak HSS v případě změny dat zašle příkaz *Push-Notification-Request* (PNR) do AS a podá detailní hlášení o těchto změnách. Odpovědí na PNR je příkaz *Push-Notification-Answer* (PNA), ten značí konec operace.

Si rozhraní

Pokud je AS tzv. CAMEL AS - Customized Applications for Mobile network Enhanced Logic Application Server (IM-SSF - Internet Protocol Multimedia Service Switching Function), tak ke komunikaci s HSS používá rozhraní Si. Toto rozhraní se

používá k přenosu CAMEL předplacených informací včetně spouštěčů z HSS do IM-SSF. Použitý protokol je zde Mobile Application Part (MAP).

Dh rozhraní

Pokud jsou v síti rozmístěny vícenásobné a samostatně adresovatelné HSS, tak AS neví, který HSS má kontaktovat. Avšak nejdříve potřebuje kontaktovat SLF. Pro tyto účely je zde zavedeno rozhraní Dh. Toto rozhraní se vždy používá ve spojení s rozhraním Sh. Použitý protokol je založen na protokolu Diameter. Jeho funkce je realizována pomocí směrovacího mechanismu prostřednictvím rozšířeného Diameter agenta. K získání adresy HSS zašle AS do SLF Sh žádost zaměřenou na HSS. Po obdržení této adresy zašle AS Sh žádost do HSS.

Rozhraní Mm

Pro komunikaci s ostatními multimediálními IP sítěmi je potřeba rozhraní mezi IMS a těmito sítěmi. Přes Mm rozhraní obdrží I-CSCF žádost o spojení z jiného SIP serveru nebo terminálu. Podobně používá S-CSCF toto rozhraní k předání IMS žádostí pocházející z UE do ostatních multimediálních sítí. Používá se zde protokol SIP.

Mg rozhraní

Toto rozhraní spojuje okrajové služby z CS - MGCF (Circuit Switched - Media Gateway Control Function) s IMS, konkrétně s I-CSCF. Přes toto rozhraní předá MGCF signalizaci příchozího spojení z CS domény do I-CSCF. Používá se zde protokol SIP. MGCF je zodpovědné za převod příchozích ISUP (ISDN User Part) signalizací na SIP signalizace.

Mi rozhraní

Jakmile S-CSCF zjistí, že má být spojení směrováno do CS domény, tak využije rozhraní Mi k předání spojení do BGCF. Používá se zde protokol SIP.

Mj rozhraní

Pokud BGCF obdrží signalizaci spojení přes Mi rozhraní, tak si vybere CS doménu, ze které tato signalizace pochází. Pokud se jedná o stejnou síť, pak se předá spojení do MGCF přes Mj rozhraní. Používá se zde protokol SIP.

Mk rozhraní

Pokud BGCF obdrží signalizaci spojení přes Mi rozhraní, tak si vybere CS doménu, ze které tato signalizace pochází. Pokud pochází z jiné sítě, pak se předá spojení do BGCF z jiné sítě přes rozhraní Mk. Používá se zde protokol SIP.

Mn rozhraní

Jde o řídící rozhraní mezi MGCF a IMS-MGW (Media Gateway Function). Řídí uživatelskou rovinu mezi IP přístupem a IMS-MGW (Mb rozhraní), dále také mezi CS přístupem (rozhraní Nb a TDM) a IMS-MGS. Mn rozhraní je založeno na standardu H.248 a je ekvivalentem k použití Mc rozhraní určeného k řízení CS-MGW.

Ut rozhraní

Jde o rozhraní mezi UE a AS. Umožňuje uživatelům bezpečně spravovat a konfigurovat jejich síťové služby související s informacemi z AS. Uživatelé mohou toto rozhraní využít k vytvoření veřejných identit služeb - Public Service Identities (PSIs), jako je seznam zdrojů a k správě autorizační politiky, která je použita danou službou. Příkladem služeb, které využívají Ut rozhraní je např. Push to talk Over Cellular a konference. AS může požadovat zajištění bezpečnosti pro toto rozhraní.

Vybraný protokol pro rozhraní Ut je Hyper Transfer Protocol (HTTP). Jakýkoli protokol vybraný k použití, který využívá rozhraní Ut, musí být založen na protokolu HTTP.

Mr rozhraní

Když potřebuje S-CSCF aktivovat služby spojené s doručitelem, tak prochází přes Mr rozhraní SIP signalizace do MRFC. Použitý protokol je SIP.

Mp rozhraní

Když potřebuje MRFC řídit nějaký mediální přenos, např. vytvořit spojení pro určitou konferenci nebo zastavit tento přenos v MRF, tak použije rozhraní Mp. Toto rozhraní plně vyhovuje standardům H.248, avšak IMS služby můžou vyžadovat určitá rozšíření.

Go rozhraní

Je v zájmu operátora, aby se zajistilo, že QoS i zdrojové a cílové adresy daného IMS mediálního toku dat odpovídají dohodnutým hodnotám na vrstvě IMS. Toto si vyžaduje komunikaci mezi IMS (řídící úroveň) a GPRS sítí (uživatelská úroveň). Nejdříve bylo toto rozhraní definováno právě pro tyto účely, později bylo přidáno účtování jako doplňková služba. Použitým protokolem je Common Open Policy Service (COPS). Procedury zde mohou být rozděleny do dvou hlavních kategorií:

Autorizace médií

- pokud jde o přístup, tak bod vyžadující politiku - Policy Enforcement Point (PEP), např. GGSN, používá Go rozhraní, aby zjistil, zda je požadovaná aktivace doručitele od PDF (Policy Decision Function) povolena. PDF zde slouží jako rozhodovací bod politiky - Policy Decision Point (PDP). PEP dále využívá toto rozhraní k informování PDP o potřebných modifikacích doručitele. Pokud jde o IMS, tak PDF využívá Go rozhraní výlučně k signalizaci, zda může nebo nemůže být doručitel použit nebo také může zažádat PEP o zahájení jeho uvolnění.

Zpoplatnění

- přes Go rozhraní je IMS schopno přenést tzv. IMS Charging Identifier (ICID), tedy identifikátor účtování, do GPRS sítě (uživatelská úroveň). Stejným způsobem může GPRS síť přenést GPRS Charging Identifier do IMS. Tímto způsobem je tedy možné později sloučit GPRS a IMS informace ohledně účtování do platebního systému.

Gq rozhraní

Pokud je nasazen samostatný PDF, tak se rozhraní Gq použije pro zaslání informací o nastavených politikách mezi použitou službou a PDF. Termín "použitou službou" zde používám, protože se předpokládá, že PDF může autorizovat i jiný provoz než jen IMS

provoz. V případě IMS zastává P-CSCF roli použitých služeb. Použitý protokol na tomto rozhraní je Diameter.

P-CSCF zašle do PDF informace o politikách každé SIP zprávy, které zahrnují SDP (Session Description Protocol). To zajišťuje, že přes PDF prochází správné informace, což umožní autorizaci médií pro všechny možné IMS relace nastavených scénářů.

P-CSCF poskytuje PDF následující informace týkající se politiky:

- Informace o mediálním toku:
  směr provozu (obousměrný, uplink, downlink),
  zdrojová/cílová IP adresa a číslo portu,
  přenosový protokol,
  maximální požadovaná šířka pásma pro uplink/downlink,
  status každého mediálního prvku (enabled/disabled pro uplink/downlink),
  informace o skupinových pravidlech těchto mediálních prvků,
  typy medií: audio, video, data, aplikace, řízení, text, zpráva.
- Zdroj rezervace politiky - obsahuje informace o tom, zda si P-CSCF přeje být kontaktován při každé autorizaci doručitele, nebo zda může PDF použít dostupné informace a rozhodnout sám.
- Oznámení o zaslání politiky - používá se k informování PDF, jestli má P-CSCF zájem přijímat oznámení o: ztrátě doručitele, obnovení doručitele, uvolnění doručitele.
- IMS identifikátor účtování (ICID) - tuto informaci dodává PDF do přístupové sítě s cílem umožnit účtování za služby.
- Informace o použitých službách - díky těmto informacím PDF rozliší QoS pro různé použité služby.
- Informace o SIP větvení: tato informace je potřeba, aby PDF správně vypočetlo autorizaci, jelikož PDF musí schválit maximální požadovanou šířku pásma jakéhokoli ze SIP dialogů, ale ne součet všech šířek pásem požadovaných všemi SIP dialogy (vyhrazení určité šířky pásma sníží kapacitu a výkon v přístupových sítích).

Navíc může P-CSCF zažádat PDF, aby odstranil předchozí autorizované zdroje. PDF používá Gq rozhraní pro doručení autorizačního tokenu, pro identifikátor účtování v GPRS, pro IP adresu GGSN a k uspokojení všech žádostí od P-CSCF zmíněných výše.

Rozhraní účtování

Rozhraní spojené s účtováním jsou Rf, Ro, Rx.

Zde je přehled všech popsaných rozhraní včetně jejich účelu a použitého protokolu:

Tab. 3-3 Souhrn rozhraní

rozhraní	spojení s objekty	účel	protokol
Gm	UE, P-CSCF	výměna informací mezi UE a CSCF	SIP
Mw	P-CSCF, I-CSCF, S-CSCF	výměna informací mezi jednotlivými CSCF	SIP
ISC	S-CSCF, I-CSCF, AS	výměna informací mezi CSCF a AS	SIP
Cx	I-CSCF, S-CSCF, HSS	komunikaci mezi I-CSCF/S-CSCF a HSS	Diameter
Dx	I-CSCF, S-CSCF, SLF	využívá ho I-CSCF/S-CSCF pro nalezení správného HSS u multi-HSS systémů	Diameter
Sh	SIP AS, OSA SCS, HSS	výměna informací mezi SIP AS/OSA SCS a HSS	Diameter
Si	IM-SSF, HSS	výměna informací mezi IM-SFF a HSS	MAP
Dh	SIP AS, OSA, SCF, IM-SFF, HSS	využívá ho AS pro nalezení správného HSS u multi-HSS systémů	Diameter
Mm	I-CSCF, S-CSCF, externí IP sítě	bude použito pro výměnu informací mezi IMS a externí sítí	Nespecifi-kován
Mg	MGCF –> I-CSCF	MGFC převádí ISUP signalizaci na SIP signalizaci a předává ji do I-CSCF	SIP
Mi	S-CSCF >BGCF	výměna informací mezi S-CSCF a BGCF	SIP
Mj	BGCF–>MGCF	výměna informací mezi BGCF a MGCF ve stejné IMS síti	SIP
Mk	BGCF–>BGCF	výměna informací mezi jednotlivými BGCF z různých IMS sítí	SIP
Mr	S-CSCF, MRFC	výměna informací mezi S-CSCF a MRFC	SIP
Mp	MRFC, MRFP	výměna informací mezi MRFC a MRFP	H.248
Mn	MGCF, MRFP	umožňuje kontrolu nad prostředky z uživatelské úrovně	H.248
Ut	UE, AS (SIP AS, OCA SCS, IM-SFF)	opravňuje UE k řízení informací spojenými s jeho službami	HTTP
Go	PDF, GGSN	umožňuje operátorům dohlížet nad QoS na uživatelské úrovni a vyměňovat informace spojené s účtováním mezi IMS a GPRS sítí	COPS

Gq	P-CSCF, PDF	výměna informací o nastavení politiky mezi P-CSCF a PDF	Diameter
Ro	AS, MRFC, S-CSCF, OCS	využívá ho AS/MRFC/S-CSCF pro online účtování pro OCS	Diameter
Rf	P-CSCF, S-CSCF, I-CSCF, BGCF, MGCF, AS, MRFC, CDF	využívají ho IMS entity pro offline účtování pro CDF (Charging Data Function)	Diameter
Rx	P-CSCF, AS, pravidla pro služby účtování	umožňuje výměnu dynamických informací o službách účtování mezi CRF (Charging Rules Function) a IMS entitami. Tyto informace využívá CRF pro výběr a dokončení pravidel pro účtování	Diameter

4 Bezpečnost v IMS

Bezpečnost je v IMS rozdělena na zabezpečení přístupu (access security) a síťovou bezpečnost (network security). Zabezpečení přístupu zahrnuje autentizaci uživatelů i sítě a také zabezpečení přenosu mezi IMS terminálem a sítí. Síťová bezpečnost se zabývá ochranou přenosu mezi síťovými uzly. Tyto uzly mohou, ale nemusí spadat pod stejného operátora.

Jak pro zabezpečení přístupu, tak i pro síťovou bezpečnost se používá IPsec (Internet Protocol Security). Avšak jádro protokolu SIP (Session Initiation Protocol) poskytuje pouze nativní podporu pro TLS (Transport Layer Security), což je nejpoužívanější bezpečnostní mechanismus pro SIP na veřejném internetu.

[1], [2], [6]

4.1 Zabezpečení přístupu

Uživatel, který se chce připojit do sítě IMS, musí nejprve podstoupit autentizaci (ověření identity uživatele služeb) a autorizaci (přidělení oprávnění přístupu k určitým službám). Jakmile je uživatel autorizován, tak probíhá SIP komunikace mezi IMS terminálem a P-CSCF pomocí dvou IPsec zabezpečených spojení.

Autentizace a autorizace uživatele a vytvoření IPsec spojení se provede pomocí zprávy REGISTER. S-CSCF s použitím autentizačních vektorů získaných z HSS (Home Subscriber Server) autentizuje a autorizuje uživatele, zatímco P-CSCF vytvoří IPsec zabezpečené spojení s terminálem. Uživatel během procesu autentizace také autentizuje síť, aby měl jistotu, že nekomunikuje s falešnou sítí.

[1], [2], [6]

4.1.1 Autentizace a autorizace

Autentizace a autorizace v IMS spoléhá na bezpečnostní funkce obsažené v IMS terminálu. Tyto funkce nejsou obsaženy přímo v IMS terminálu, ale jsou na kartě, která je vložena do tohoto terminálu.

V 3GPP (The 3rd Generation Partnership Project) sítích se tato karta nazývá UICC (Universal Integrated Circuit Card). UICC může obsahovat jednu nebo více aplikací, jako na

Obr. 4-1. Každá aplikace si ukládá do paměti několik konfigurací a parametrů vztaženým k jednotlivým užitím. Těmito aplikacemi jsou:

SIM (Subscriber Identity Module)

SIM obsahuje paměť pro předplacené informace, uživatelské předvolby, autentizační klíč, zprávy atd. a je nezbytná pro provoz terminálů v GSM síti. Ačkoli se někdy termíny UICC a SIM zaměňují, tak UICC označuje fyzickou kartu, kdežto SIM označuje jednu aplikaci z UICC. SIM se hodně využívá v 2G (Second Generation) sítích - GSM sítích.

USIM (UMTS Subscriber Identity Module)

Jde o další příklad aplikace, která vychází z třetí generace UICC. Obsahuje další skupinu parametrů: předplacené informace, autentizační informace, způsoby placení a

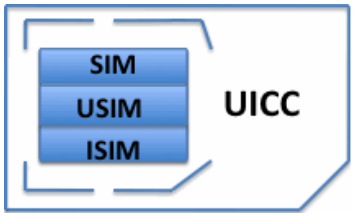
paměť pro zprávy. Používá se v 3G (Third Generation) sítích - UMTS sítích. SIM a USIM spolu samozřejmě můžou existovat v jedné UICC. Pokud je k tomu terminál uzpůsobený, tak může využívat jak GSM tak UMTS sítě.

ISIM (IP-Multimedia Services Identity Module)

Je mimořádně důležitá pro IMS, protože obsahuje parametry pro identifikaci uživatele, autentizaci a konfiguraci terminálu. ISIM může existovat spolu se SIM či USIM nebo s oběma aplikacemi na stejné UICC.

3GPP sítě povolí přístup do IMS, pokud UICC obsahuje ISIM nebo USIM. ISIM je sice preferovanější, ale v případě, že uživatel má starší kartu UICC, která ISIM neobsahuje, tak má přístup i s USIM. Přístup do IMS se SIM není kvůli slabému zabezpečení povolen.

Identifikační parametry které byly v 3GPP IMS uloženy v ISIM jsou v 3GPP2 IMS uloženy v IMS terminálu nebo v R-UIM (Removable User Identity Module). Tyto parametry jsou v obou sítích stejné, stejně tak jako bezpečnostní funkce.



Obr. 4-1 Karta UICC

4.1.2 Autentizace a autorizace s ISIM

Nyní popíši autentizační a autorizační procedury, které se odehrávají mezi IMS terminálem s ISIM aplikací a sítí.

Vzájemná autentizace mezi uživatelem a sítí v IMS je založena na dlouhodobě sdíleném tajemství mezi ISIM v terminálu a HSS v síti. Každá ISIM obsahuje tajný klíč. Tajný klíč od každé ISIM je také uložen v jejich domovském HSS. K vytvoření vzájemné autentizace si musí ISIM s HSS navzájem dokázat, že znají tajný klíč. Avšak terminál, který obsahuje ISIM, komunikuje pomocí SIP, zatímco HSS nikoliv. Aby se tento problém vyřešil, tak S-CSCF, přiřazený k uživateli, převezme roli autentizátora. HSS tak efektivně přenese tuto roli na S-CSCF.

S-CSCF k získání autentizačních vektorů z HSS a k vyzvání uživatele (UE) používá protokol Diameter. Tyto autentizační vektory obsahují výzvu a očekávanou odpověď na tuto výzvu od uživatele. Pokud uživatel odpoví odlišně, považuje S-CSCF tuto autentizaci za neúspěšnou.

Nyní popíši jak S-CSCF namapuje tyto výzvy do zprávy *REGISTER*. První co IMS terminál udělá při přihlašování do IMS sítě je, že pošle žádost *REGISTER* do své domovské sítě. I-CSCF zajišťuje přiřazení zprávy *REGISTER* k S-CSCF pro uživatele, po kterém je požadována jeho autentizace a autorizace. Postupuje podle kritérií získaných z HSS - výměna zpráv pomocí Diameter protokolu (3) a (4). S-CSCF musí stáhnout čísla autentizačních vektorů z HSS (7). Každý vektor obsahuje náhodnou výzvu (RAND), autentizační token (AUTN), očekávanou odpověď od IMS terminálu (XRES), klíč pro ověření integrity (IK) a klíč pro šifrování (CK). HSS vytvoří AUTN pomocí tajného klíče, který sdílí s ISIM a s pomocí sekvence čísel (SQN), která je udržována mezi ISIM

a HSS. Každý autentizační vektor může být k autentizaci ISIM použit pouze jednou. S-CSCF stáhne několik vektorů, aby se zabránilo tomu, že se bude kontaktovat HSS pokaždé, když bude chtít uživatele znovu autentizovat.

S-CSCF použije první autentizační vektor k vytvoření shrnutí výzvy pro ISIM. S-CSCF pošle odpověď *401 Unauthorized* (8) (neautorizován). Ta obsahuje v hlavičce pole *WWW-Authenticate*. Zde je zakódovaný (64bit) AUTN a RAND, použitý algoritmus je AKAv1-MD5.

Pole *WWW-Authenticate*:

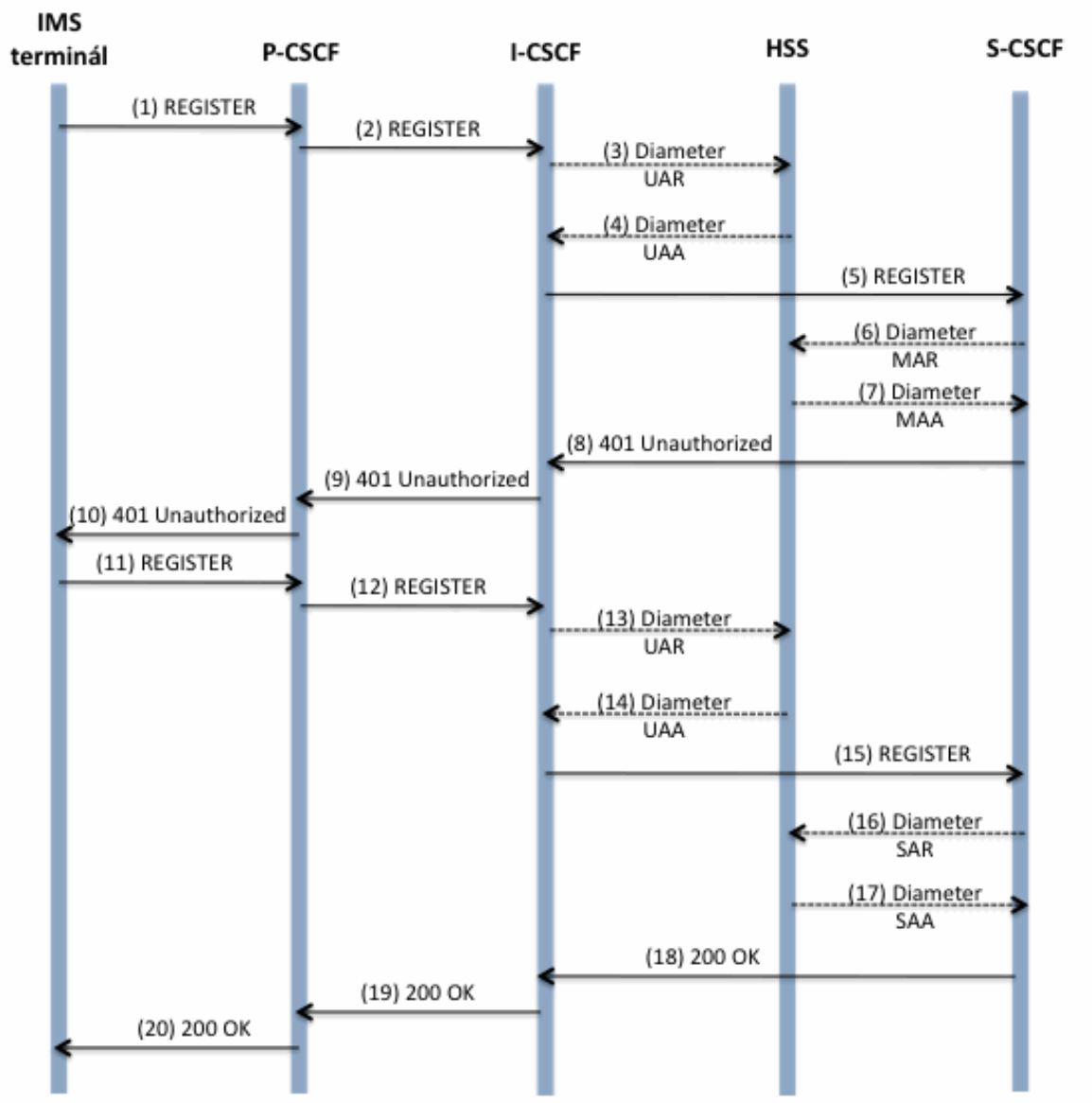
```
WWW-Authenticate: Digest
realm="domain.com",
nonce="CjPk9mRqNuT25eRkajM09uT19nM09uT19nMz50X25PZz==",
qop="auth, auth-int",
algorithm=AKAv1-MD5
```

Jakmile obdrží terminál odpověď *401 Unauthorized* (10), tak odvodí RAND, AUTN a klíče CK a IK z hodnoty v řádku "*nonce*". ISIM pak vypočítá AUTN pomocí SQN a svého tajného klíče. Pokud získá stejný AUTN jaký obdržel, tak považuje tuto síť za autentizovanou. V tomto případě používá ISIM svůj vlastní tajný klíč a přijatou náhodnou výzvu RAND k vygenerování odpovědi (RES), která se vrací do S-CSCF v poli hlavičky *Authorization* nové žádosti *REGISTER* (11).

Ukázka hlavičky nyní:

```
WWW-Authenticate: Digest
username="Jan.Novak@domain.com",
realm="domain.com",
nonce="CjPk9mRqNuT25eRkajM09uT19nM09uT19nMz50X25PZz==",
uri="sip:domain.com",
qop="auth-int",
nc=00000001,
cnonce="0a4f113b",
response="6629fae49393a05397450978507c4ef1",
```

Jakmile S-CSCF obdrží tuto žádost *REGISTER* (15), tak porovná hodnotu přijaté RES s očekávanou hodnotou XRES. Pokud se shodují, tak S-CSCF považuje uživatele za autentizovaného a vrací odpověď *200 OK* (18). Všechny tyto procedury jsou zakresleny na Obr. 4-2.



Obr. 4-2 Průběh počáteční registrace

4.1.3 Autentizace a autorizace s USIM

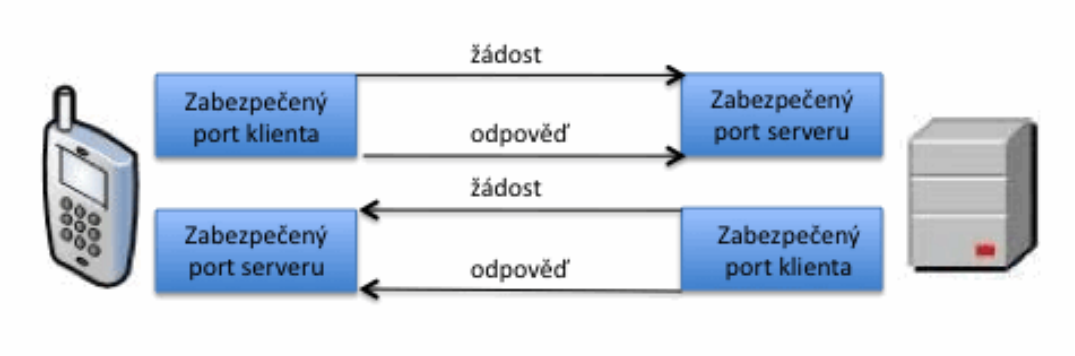
IMS terminál, který obsahuje UICC s USIM, ale bez ISIM, může stále využívat IMS síť. USIM samozřejmě neobsahuje soukromé a veřejné identity uživatelů, ani dlouhodobé heslo potřebné pro autentizaci uživatele v IMS síti. Pořád ale ještě obsahuje IMSI (ekvivalent pro soukromou identitu uživatele v sítích s přepínáním okruhů/paketů). IMS terminál vytvoří dočasnou soukromou i dočasnou veřejnou identitu uživatele a domovskou doménu URI na obsah IMSI. USIM dále obsahuje dlouhodobé heslo typicky používané pro autentizaci uživatele v sítích s přepínáním okruhů a v sítích s přepínáním paketů. Pokud je USIM použit k připojení do IMS sítě, tak se toto heslo použije k autentizačním účelům, jak ze strany sítě (uloženo v HSS), tak ze strany terminálu (uloženo v USIM).

Ve většině případů nebude chtít domácí operátor zveřejnit buď IMSI nebo soukromou identitu uživatele mimo domovskou síť. Dočasné veřejné i soukromé identity uživatelů však pochází z IMSI a jsou viditelné ve zprávách SIP. Proto může domácí

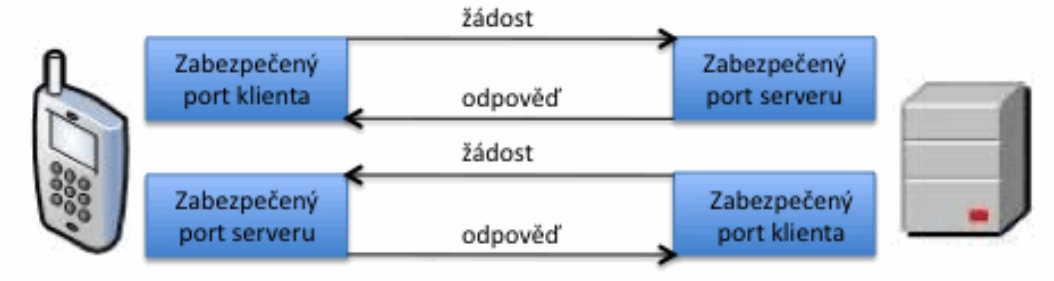
operátor zabránit jakékoliv veřejné identitě uživatele, tedy i té dočasné, aby se používala ve zprávách SIP jinak, než žádost REGISTER a odpověď na ni. IMS terminál může použít jakoukoli z veřejných identit uživatele, které jsou tomuto uživateli přiděleny, tak jak jsou přeneseny do terminálu v P-Associated-URI hlavičce u odpovědi *200 OK* na žádost *REGISTER*. Pokud IMS terminál zahájí spojení s touto zakázanou veřejnou identitou uživatele, tak S-CSCF zamítne ustálení spojení.

4.1.4 Ustálení zabezpečeného spojení

P-CSCF a terminál mezi sebou založí dvě IPsec zabezpečené spojení. Použitím dvou zabezpečených spojení namísto jednoho dovolí terminálům a P-CSCF používat UDP (Obr. 4-3) pro příjem odpovědí na žádosti na jiném portu, než na kterém odesílají tyto žádosti. Na druhou stranu ale terminály a P-CSCF využívající TCP (Obr. 4-4) posílají odpovědi po stejném TCP spojení, tedy na stejném portu, na jakém obdrží žádost.



Obr. 4-3 Využití portů a zabezpečené spojení s UDP



Obr. 4-4 Využití portů a zabezpečené spojení s TCP

P-CSCF a terminál se musí shodnout na nastavení parametrů pro ustálení dvou IP sec zabezpečených spojení mezi sebou. P-CSCF obdrží klíč integrity (IK) a šifrovací klíč (CK) v odpovědi *401 Unauthorized* od S-CSCF (ten je dostal v autentizačním vektoru od HSS). P-CSCF odstraní z této odpovědi oba klíče (IK a CK) ještě před odesláním do IMS terminálu. P-CSCF i IMS terminál používají dvě stejné zprávy *REGISTER* (Obr. 4-2), které jsou použity pro autentizaci, k dohodnutí zbývajících IPsec parametrů.

Terminál přidá pole *Security-Client* do hlavičky *REGISTER* (1). Toto pole obsahuje mechanismy (*ipsec-3gpp*) a algoritmy (*hmac-sha-1-96*), které terminál podporuje, SPI (Security Parameter Index) - identifikátory zabezpečeného spojení a čísla portů, které budou použity.

Pole *Security-Client*:

```
Security-Client:    ipsec-3gpp; alg=hmac-sha-1-96;
                    spi-c=23456789; spi-s=12345678;
                    port-c=2468; port-s=1357;
```

P-CSCF přidá pole *Security-Server* do hlavičky odpovědi *401 Unauthorized* (10). Toto pole obsahuje mechanismy (*ipsec-3gpp*) a algoritmy (*hmac-sha-1-96*), které P-CSCF podporuje, SPI identifikátory zabezpečeného spojení a čísla portů, které budou použity.

Pole *Security-Server*:

```
Security-Client:    ipsec-3gpp;q=0.1;alg=hmac-sha-1-96;
                    spi-c=98765432; spi-s=87654321;
                    port-c=8642; port-s=7531;
```

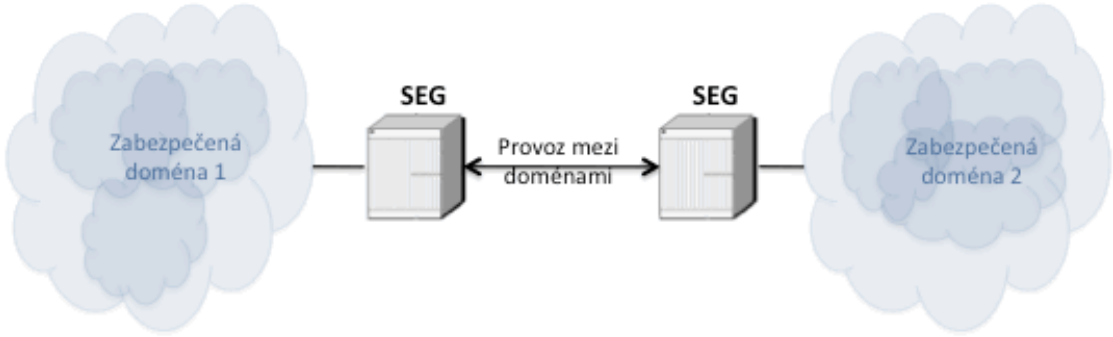
V tomto případě obsahuje pole *Security-Server* pouze jeden mechanismus. Pokud by jich však obsahoval více, tak bude preferován mechanismus s vyšším číslem *q*.

Zabezpečené spojení je připraveno k použití, jakmile terminál obdrží pole *Security-Server* v hlavičce odpovědi *401 Unauthorized*. Terminál tedy pošle žádost *REGISTER* (11) přes jedno z právě ustálených zabezpečených spojení. Terminál obsahující pole *Security-Verify* v hlavičce této žádosti *REGISTER* duplikuje obsah pole *Security-Server* z předchozí obdržené hlavičky. Tímto se server ujistí, že tu není nikdo mezi nimi, kdo by mohl modifikovat seznam bezpečnostních mechanismů, který je poslán klientovi. Útočník by totiž mohl z tohoto seznamu odstranit silnější bezpečnostní mechanismy a nahradit je slabšími, přes které by snadněji pronikl. Útočník, který by modifikoval seznam *Security-Server*, by musel prolomit bezpečnostní mechanismus vybraný v reálném čase, aby také modifikoval pole *Security-Verify* v hlavičce. Jinak by P-CSCF odhalil útok a zrušil registraci. Tento výběr bezpečnostních mechanismů bude bezpečný, dokud nebude možné prolomit nejslabší mechanismus na seznamu v reálném čase.

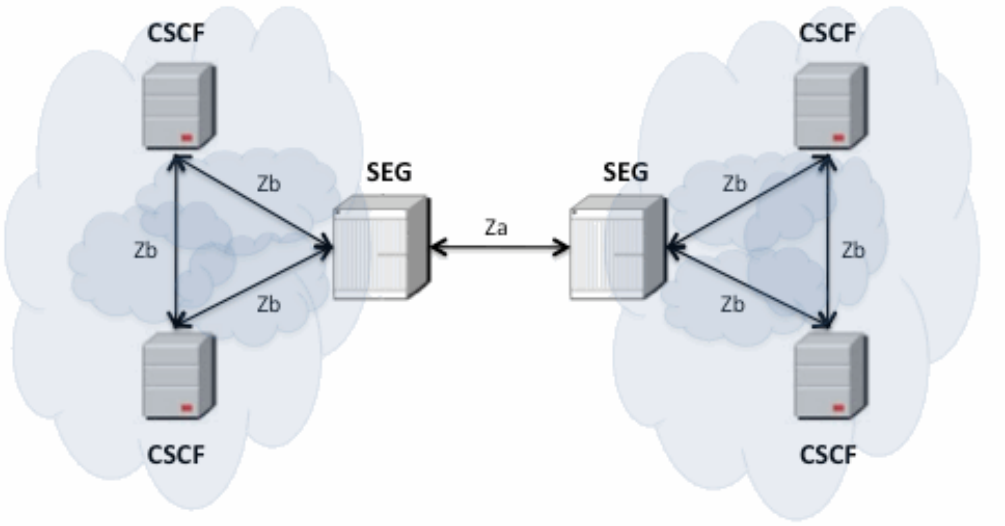
4.2 Síťová bezpečnost

Síťová bezpečnost se zabývá zabezpečením provozu mezi odlišnými zabezpečenými doménami. Zabezpečenou doménou se zde myslí síť, která je řízená jedinou administrativní autoritou. Např. spojení, kde P-CSCF a S-CSCF jsou umístěny v odlišných sítích představuje provoz mezi různými doménami.

[1], [2], [6]



Obr. 4-5 Provoz mezi doménami skrze dvě zabezpečené brány



Obr. 4-6 Rozhraní Za a Zb

Veškerý příchozí nebo odchozí provoz ze zabezpečené domény prochází přes bránu SEG (Security Gateway). Z toho tedy plyne, že provoz z jedné domény do druhé musí procházet přes dvě brány SEG, jak je vidět na Obr. 4-5.

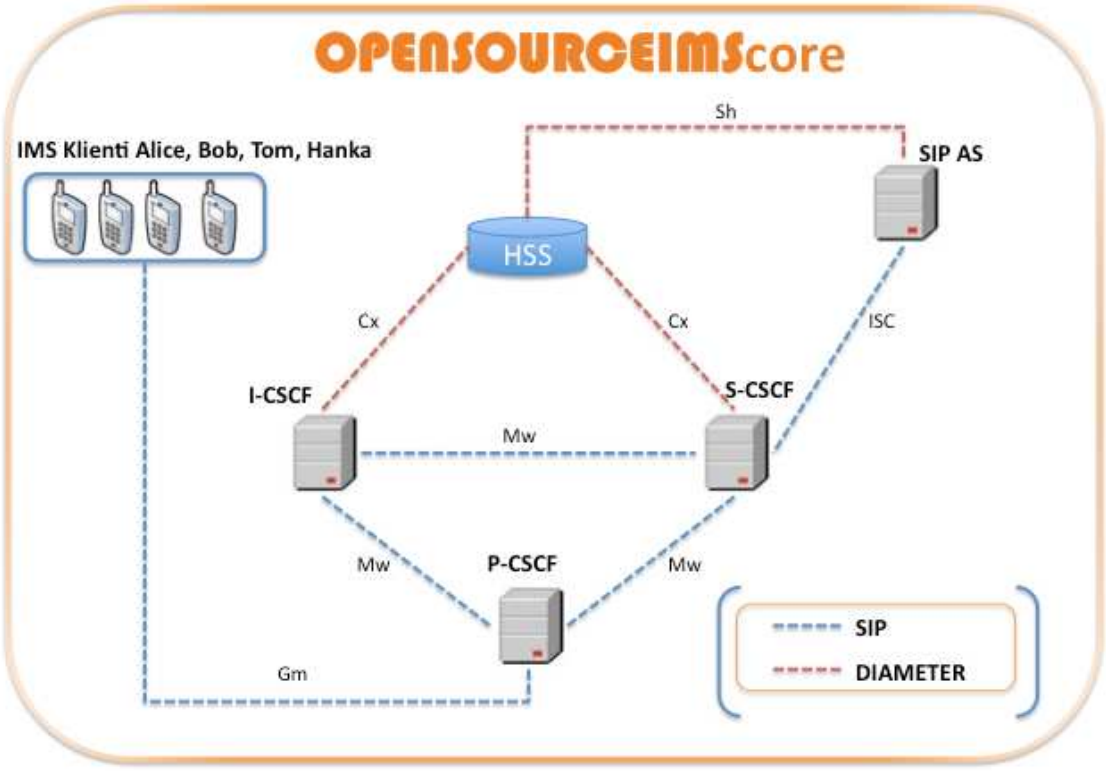
Provoz mezi jednotlivými bránami SEG je zabezpečen pomocí IPsec ESP (Encapsulated Security Payload), který je tunelován. Zabezpečené spojení mezi bránami SEG je ustáleno a udržováno pomocí IKE (Internet Key Exchange) - výměny klíče po internetu. Uvnitř zabezpečené domény komunikují síťové entity s bránami SEG pomocí IPsec. Toto zabezpečené spojení používají také ke komunikaci mezi sebou. Rozhraní mezi normálními síťovými entitami a mezi síťovými entitami a bránami SEG je stejné a označuje se jako *Zb*. Rozhraní mezi bránami SEG jiných domén se označuje jako *Za*, viz. Obr. 4-6.

Na rozhraní *Za* je povinná Autentizace, ochrana integrity a šifrování. To poskytuje provozu mezi IMS doménami maximální stupeň ochrany. Rozhraní *Zb* je navrženo pro ochranu IMS na signalizační úrovni. Jelikož přes toto rozhraní prochází jen provoz "uvnitř operátora", tak je jen na operátorovi, zda ho rozmístí, a v případě že ano, tak jaké bezpečnostní služby zahrne. Ochrana integrity je v případě implementace *Zb* rozhraní povinná, ale šifrování je volitelné.

5 Open IMS Core

Open IMS Core představuje open source projekt Fraunhoferova Institutu FOKUS v Berlíně, jenž poskytuje základní implementaci pro testování IMS technologie a vývoj IMS aplikací. Je založen na linuxovém jádře, takže je jeho instalace možná jen na tyto systémy. Je tvořen SIP servery CSCF (I-CSCF, S-CSCF a P-CSCF) a jednoduchou databází uživatelů HSS, což je základ každé IMS architektury. Architekturu tohoto prostředí můžeme vidět na Obr. 5-1.

[10]



Obr. 5-1 Architektura Open IMS Core prostředí

Servery CSCF zde působí jako hlavní směrovací prvky pro jakoukoli IMS signalizaci. Jsou postaveny na SER (SIP Express Router), který může působit jako SIP registrátor, proxy nebo redirect server a je schopen zvládnout tisíce hovorů za vteřinu.

Jako databázi uživatelů HSS zde FOKUS vyvinul svou vlastní označovanou jako FHoSS (FOKUS Home Subscriber Server). Ta je kompletně napsána v jazyce Java a uživatelské informace jsou uloženy v databázi MySQL. Jelikož slouží jako databáze, tak se zaměřuje více než na výkon na pravdivost a shodu uložených informací.

5.1 Instalace

Instalace Open IMS Core na linuxový operační systém Ubuntu 8.10 Intrepid Ibex. [12]

I. Základní požadavky

Hardware

Počítač se systémem Linux.

Software

- 100MB volného místa na disku,
- GCC (GNU Compiler Collection) 3/4,
- utilita make a ANT,
- JDK (Java Development Kit) verze 1.5 a vyšší,
- MySQL nainstalováno a spuštěno (nebo jiný databázový systém),
- víceúčelový generátor syntaktických analyzátorů bison a utilitu flex,
- XML knihovnu libxml2 (> 2.6) a MySQL knihovnu libmysql,
- Linuxové jádro 2.6 a ipsec-tools pro volbu zabezpečení IPsec,
- volitelně: openssl pro případné využití zabezpečené komunikace pomocí TLS (Transport Layer Security),
- DNS server BIND nainstalován a spuštěn,
- internetový prohlížeč,
- vše předchozí se požaduje nainstalováno, nakonfigurováno a spuštěno.

Použité příkazy k instalaci potřebných balíčků:

```
sudo apt-get install gcc
sudo apt-get install ant
sudo apt-get install sun-java6-jdk
sudo apt-get install mysql-server
sudo apt-get install flex
sudo apt-get install bison
sudo apt-get install libxml2-dev libmysqlclient15-dev
sudo apt-get install bind9
```

II. Stáhnutí zdrojového kódu

Nejprve nainstalujeme Subversion (SVN) pro následné získání pracovní kopie.

```
sudo apt-get install subversion
```

Vytvoříme pracovní adresář pro zdrojový kód, změníme vlastníka (username) a přesuneme se do tohoto adresáře.

```
mkdir /opt/OpenIMSCore
sudo  chown -R username /opt/OpenIMSCore/
cd /opt/OpenIMSCore
```

Zde vytvoříme adresář *ser_ims* a do něj stáhneme pracovní kopii serverů CSCF.

```
mkdir ser_ims
svn checkout http://svn.berlios.de/svnroot/repos/
openimscore/ ser_ims/trunk ser_ims
```

Vytvoříme adresář FHoSS a do něj stáhneme pracovní kopii serveru HSS.

```
mkdir FHoSS
svn checkout http://svn.berlios.de/svnroot/repos/
openimscore/ FHoSS/trunk FHoSS
```

Pokud v předchozích krocích vybereme jiné cesty, tak nesmíme zapomenout změnit příslušné konfigurační soubory.

III. Kompilace

Kompilace Ser_ims

```
cd ser_ims
make install-libs all
```

Pokud by v tomto kroku došlo k chybě, pravděpodobně nemáme splněny některé ze základních softwarových požadavků. V mém případě proběhlo vše v pořádku, takže pokračuji následnou kompilací FHoSS.

Kompilace FHoSS

Zde se nejprve ujistíme, že opravdu používáme JDK verze 1.5 a vyšší.

```
# java -version
java version "1.5.0_07"
Java(TM) 2 Runtime Environment, Standard Edition(build
1.5.0_07-b03)
Java HotSpot(TM) Client VM (build 1.5.0_07-b03, mixed mode)
```

Z výpisu je vidět, že naše verze JDK vyhovuje požadavkům a můžeme tedy přejít k samotné kompilaci.

```
cd FHoSS
ant compile
ant deploy
```

IV. Konfigurace systémového prostředí

Výchozí nastavení odpovídá práci na lokální smyčce (*127.0.0.1*) a přednastavená doména je *open-ims.test*. Přístupová práva k MySQL jsou nastavena pouze pro lokální přístup. Změnu nastavení provádíme v konfiguračním souboru *ser_ims/cfg/configurator.sh*:

- přepsáním *127.0.0.1* vlastní IP adresou,
- přepsáním domácí domény (*open-ims.test*) vlastní doménou,
- změnou MySQL hesla.

DNS

Zde editujeme konfigurační soubor /etc/dhcp3/dhclient.conf, kde odkomentujeme řádek *prepend domain_name_servers 127.0.0.1*. Ukázkový DNS konfigurační soubor nalezneme v ser_ims/cfg/open-ims.dnszone a zkopírujeme jej do adresáře DNS serveru BIND.

```
sudo cp /opt/OpenIMSCore/ser_ims/cfg/open-ims.dnszone/
etc/bind/
```

Změníme práva:

```
chown -R named:named /etc/bind/open-ims.dnszone.
```

Upravíme konfigurační soubor DNS serveru Bind. Tento soubor najdeme v */etc/Bind/named.conf*. Zde dopíšeme následující nastavení: (IP adresu DNS serveru najdeme v */etc/resolv.conf*)

```
options {
      ...
      forward first;
      forwarders {
            {IP adresa upstream DNS serveru;}  ;
      };
      ...
};
...
zone "open-ims.test" IN {
      type master;
      file "pri/open-ims.dnszone";
      notify no;
```

Nyní restartujeme DNS server Bind.

```
sudo /etc/init.d/bind9 restart
```

V tomto kroku můžeme vyzkoušet, jestli vše pracuje tak jak má pomocí následujícího příkazu:

```
dig @127.0.0.1 pcscf.open-ims.test.
```

Abychom mohli DNS server používat, bude náš soubor /etc/resolv.conf vypadat následovně:

```
# cat /etc/resolv.conf
nameserver 127.0.0.1
search open-ims.test
domain open-ims.test
```

Musíme mít na paměti, že utility jako DHCP klient nám může tento soubor přepsat do výchozího nastavení.

MySQL

Zde provedeme nastavení databází pro I-CSCF, HSS a uživatelská data.

```
mysql -u root -p -h localhost < dump.sql
mysql -u root -p -h localhost < ser_ims/cfg/icscf.sql
mysql -u root -p -h localhost < FHoSS/scripts/hss_db.sql
mysql -u root -p -h localhost < FHoSS/scripts/userdata.sql
```

Nyní už jen zkontrolujeme, že databáze jsou na správném místě a jsou dostupné.

V. Konfigurace IMS Core

CSCF

Zkopírujeme následující soubory do */opt/OpenIMSCore*.

```
cp ser_ims/cfg/*.cfg .
cp ser_ims/cfg/*.xml .
cp ser_ims/cfg/*.sh .
```

FHoSS

V adresáři *opt/OpenIMSCore/FHoSS/deploy* najdeme konfigurační soubory, nesmíme zapomenout na update *open-ims.dnszone* souboru a restartovat DNS server.

VI. Spuštění jednotlivých částí OpenIMSCore

CSCF

Spustíme jednotlivé servery CSCF. Všechny tyto procesy by měly běžet paralelně a v jednotlivých oknech pak uvidíme pravidelný výpis logů těchto procesů.

```
cd /opt/OpenIMSCore
./pcscf.sh
./icscf.sh
./scscf.sh
```

FHoSS

Spustíme server s databází uživatelů.

```
cd /opt/OpenIMSCore/FHoSS/deploy
./startup.sh
```

Pokud by se předchozí krok nezdařil, bude nejspíše problém s proměnou *JAVA_HOME* a s modifikací spouštěného skriptu. Nyní můžeme zkontrolovat i webové rozhraní pro přístup k HSS na adrese *http://localhost:8080* s následujícími přihlašovacími údaji - *login: hssAdmin password: hss*. Zkontrolujeme, že servery I-CSCF a S-CSCF jsou v pořádku připojeny k databázovému serveru HSS.

VI. Konfigurace účastníků

FHoSS

Zde nalezneme dva předem nakonfigurované účty *alice@open-ims.test* a *bob@open-ims.test*. Pro vytvoření nového uživatele jsou potřeba následující kroky:

- vytvoříme uživatele,
- vytvoříme soukromou identitu,
- vytvoříme veřejnou identitu,
- spojit.

Detailní postup pro vytvoření nového účtu popíši v následující kapitole.

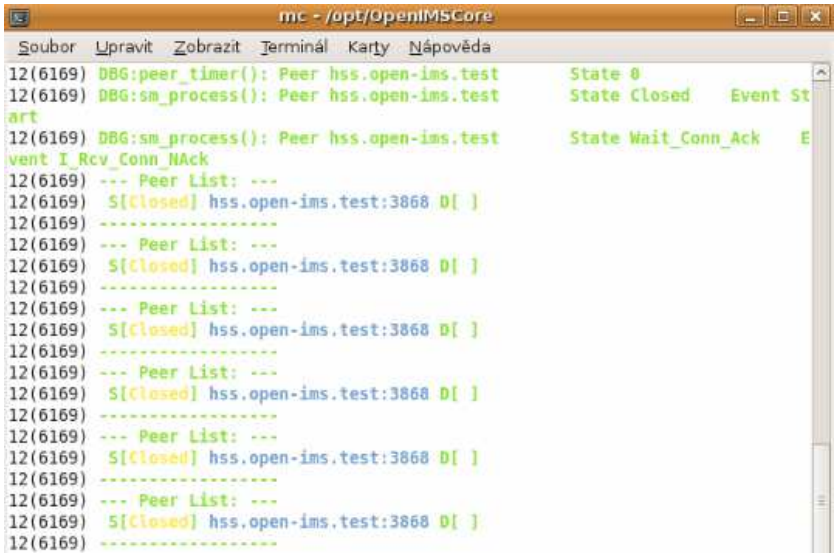
VII. Ověření funkčnosti

Nyní už máme vše nainstalováno a nakonfigurováno, a tak můžeme vyzkoušet všechny potřebné komponenty a jejich správné nastavení. K tomu nám poslouží registrace jednoho z vytvořených účtů, která využívá všechny součásti IMS sítě a je tak dobrým předpokladem pro ověření správnosti nastavení všech potřebných prvků.

V případě výskytu jakýchkoliv problémů můžeme využít síťový analyzátor Wireshark, který nám pomůže pomocí analýzy jednotlivých paketů nalézt problém v naší sestavené síti. Pro SIP komunikaci se zaměříme na porty 4060, 5060 a 6060. Pro komunikaci Diameter protokolem pak porty 3868, 3869 a 3870.

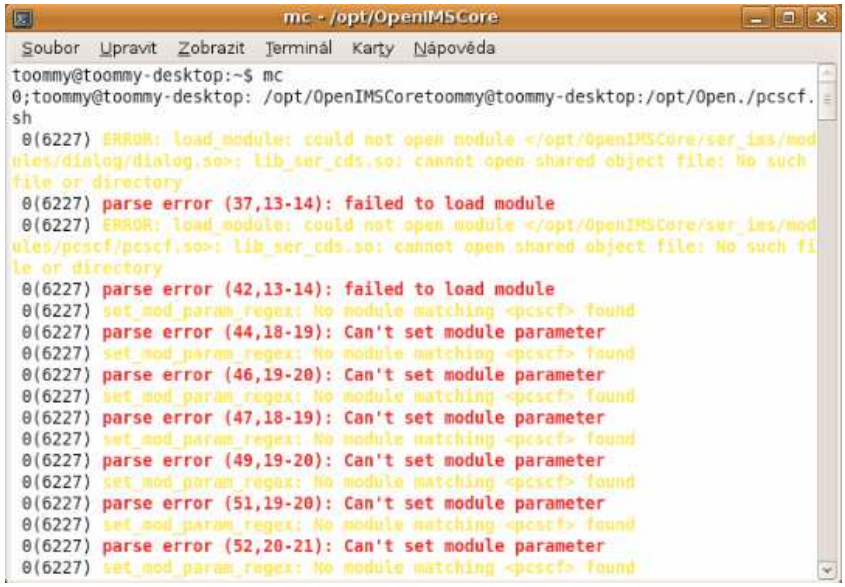
5.2 Komplikace v Ubuntu

Veškerá instalace systému OpenIMSCore proběhla dle návodu na [10] pod OS Ubuntu 8.10 Intrepid Ibex v pořádku. Přesto však nastaly v poslední fázi při spouštění jednotlivých CSCF serverů a databázového serveru HSS problémy. Na Obr. 5-2 vidíme spouštění skriptu icscf.sh, jenž by měl spustit server I-CSCF. Tento skript však hlásí chybu ve spojení s HSS.



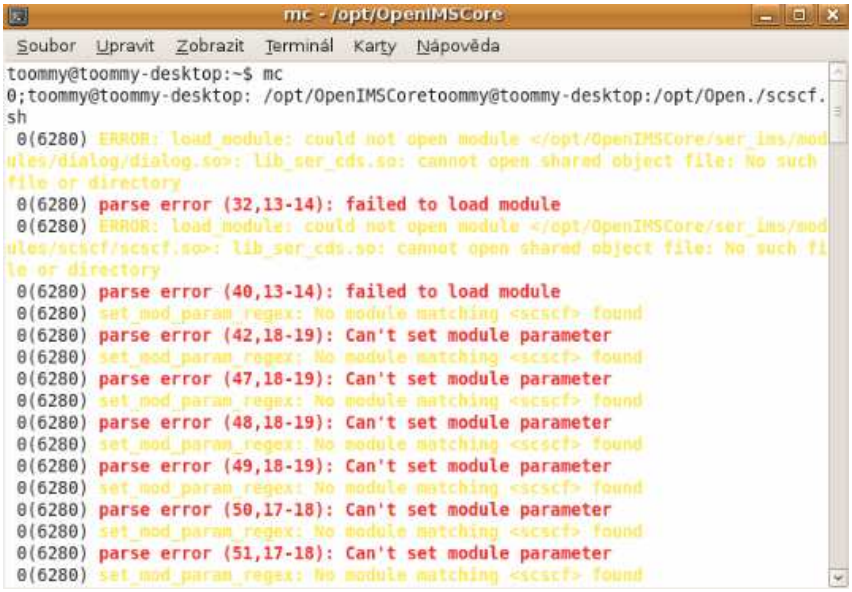
Obr. 5-2 Chyba I-CSCF

Další chyba nastala u skriptu pcscf.sh, jenž by měl spustit server P-CSCF. Ten není schopen zavézt patřičné moduly, jak vidíme na Obr. 5-3.



Obr. 5-3 Chyba P-CSCF

Stejná chyba jako v předchozím kroku nás čekala i u skriptu scscf.sh jenž slouží ke spuštění serveru S-CSCF. Opět není schopen zavézt patřičné moduly, jak vidíme na Obr. 5-4.



Obr. 5-4 Chyba S-CSCF

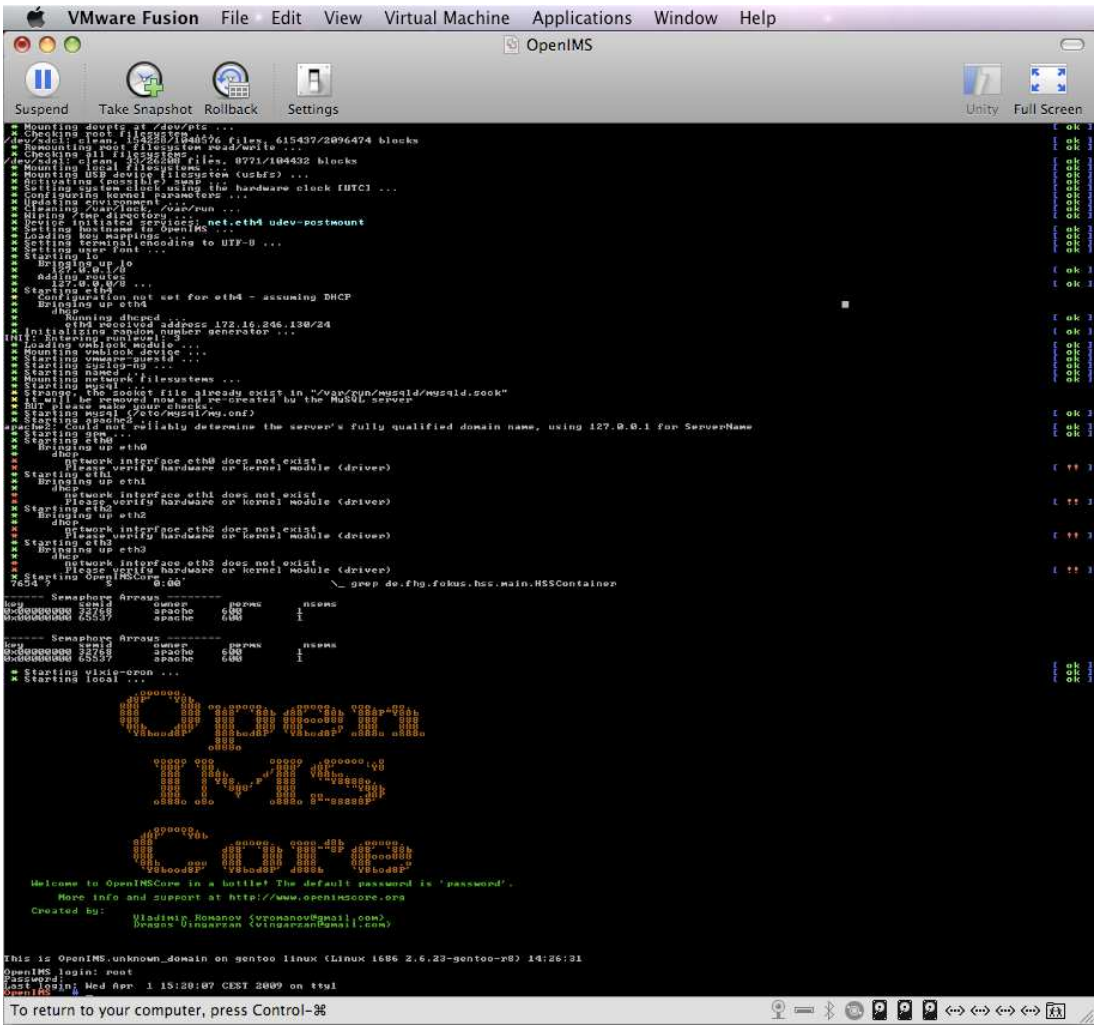
Poslední chyba nastala u skriptu startup.sh sloužícího ke spuštění HSS databázového serveru. Zde je hlášena chyba spojená s nastavením cesty k nástrojům Java, viz Obr. 5-5. Tuto chybu vyřešíme přepsáním cesty pro proměnou *JAVA_HOME/usr/lib/jvm/java-6-sun-1.6.0.10/bin/java.*

Obr. 5-5 Chyba HSS

Po zdlouhavých řešeních vyskytlých problémů, které bohužel nevedly k jejich odstranění, jsem měl na výběr dvě možnosti. Jedna z nich byla instalace systému OpenIMSCore na linuxovou distribuci Gentoo, jenž se zdá jako nejkompatibilnější pro tento systém. Vzhledem k tomu, že už jsem si celou instalací OpenIMSCore systému

prošel, a že instalace distrubuce Gentoo může trvat klidně i několik dní, tak jsem přešel k možnosti druhé. Tou se stal připravený image soubor pro spuštění v libovolném virtuálním stroji, získaný z [10]. Tento image OpenIMSCore systému běží právě na linuxové distribuci Gentoo, což znamená velkou úsporu času případné instalace této distribuce a také zaručenou kompatibilitu se systémem OpenIMSCore.

Image OpenIMSCore jsem spouštěl na virtuálním stroji VMware Fusion pod Mac OS X Leopard. Na Obr. 5-6 vidíme počáteční přihlášení do systému OpenIMSCore. Zadané přihlašovací údaje - Open IMS login: root, Password: password. Pokud nechceme pracovat v konzolovém režimu, máme možnost se přepnou do zjednodušeného grafické rozhraní TWM zadáním příkazu *startx* nebo do plně grafického rozhraní KDE pomocí příkazu */etc/init.d/xdm start*. V případě druhého příkazu však musíme mít image s tímto grafickým rozhraním KDE.



Obr. 5-6 Spuštění OpenIMSCore ve VMware

5.3 Vytvoření nových účtů

Nového uživatele v systému OpenIMSCore máme možnost vytvořit buď pomocí webového rozhraní k HSS na adrese *http://localhost:8080* nebo pomocí konfiguračního skriptu, který se nachází v */opt/OpenIMSCore/ser_ims/cfg/add-imscore-user_newdb.sh*.

Ať už si zvolíme kteroukoliv možnost, princip zůstává vždy stejný. Aby mohl být nový uživatel úspěšně přidán do databáze uživatelů v HSS a komunikovat v IMS síti, musíme pro něj vytvořit následující:

- IMSU - IMS Subscription - uživatele,
- IMPI - soukromá identita uživatele,
- IMPU - veřejná identita uživatele.

Pro vytvoření nového uživatele pomocí konfiguračního skriptu se nejdříve přepneme do adresáře, kde se tento skript nachází, tedy */opt/OpenIMSCore/ser_ims/cfg/*. Poté spustíme konfigurační skript *add-imscore-user_newdb.sh* s parametrem *-u <user>* pomocí kterého definujeme nejen jméno nového uživatele, ale zároveň i jeho IMPI, IMPU a heslo. Druhým použitým parametrem pro spouštěný skript je *-a*, který nám zajistí automatické použití námi vytvořeného skriptu pro přidání nového uživatele *add-user-hanka.sql*.

Postup pro přidání nového uživatele hanka:

```
cd /opt/OpenIMSCore/ser_ims/cfg/
./ add-imscore-user_newdb.sh -u hanka –a
```

```
add-imscore-user.sh -u <user> [-t <tel-URI>] [-r <realm> -p <password>] [-a|-d
]
  -u <user>: The username (e.g. 'brooke')
  -t <tel-URI>: The tel-URI (e.g. 'tel:+49123456789'). If used with -a, -d req
uires
     this parameter too!
  -r <realm>: The realm. Default is 'open-ims.test'
  -i <impi>: The Private Identity (e.g. 'brooke@open-ims.test'). The -u option
 overrides this.
  -b <impu>: The Public Identity (e.g. 'sip:brooke@open-ims.test') The -u opti
on overrides this.
  -p <password>: The password. Default is value of -u option
  -a: Automatically apply created add script
  -d: Automatically apply created delete script
  -c: Delete the scripts afterwards (by default they are not deleted)

OpenIMS ~ # cd /opt/OpenIMSCore/ser_ims/cfg
OpenIMS cfg # ./add-imscore-user_newdb.sh -u hanka -a
Successfully wrote add-user-hanka.sql
Successfully wrote delete-user-hanka.sql
Apply add-user-hanka.sql as user root...
Enter password:
Successfully applied add-user-hanka.sql
OpenIMS cfg #
```

Obr. 5-7 Vytvoření nového uživatele pomocí skriptu

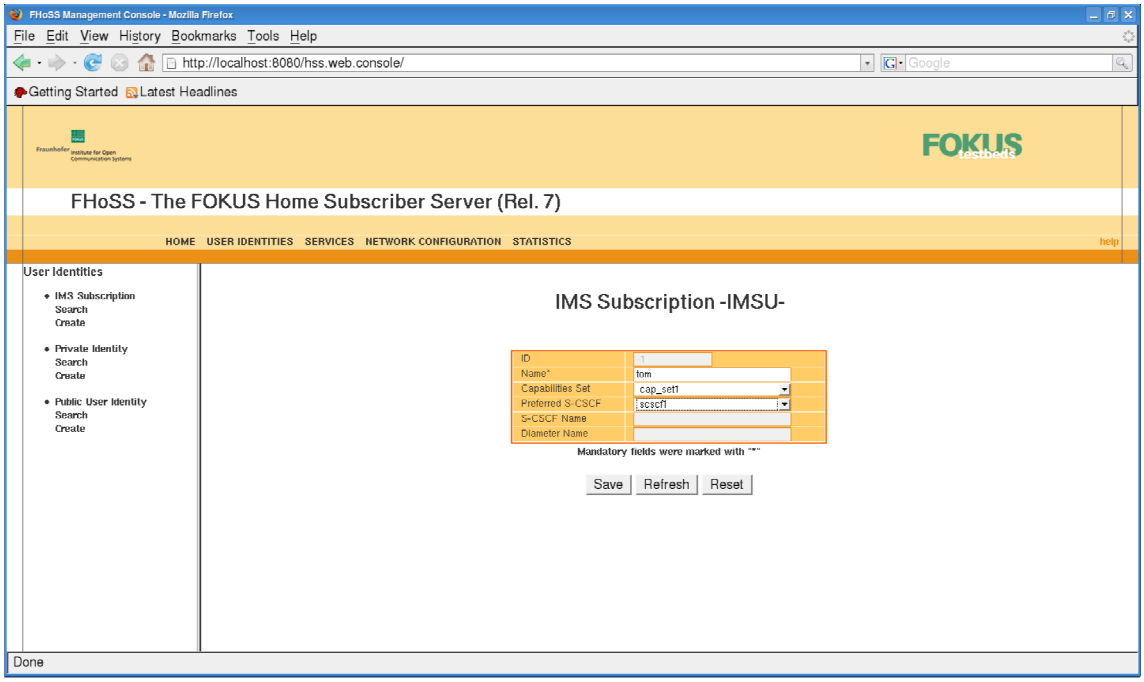
Přehled volitelných parametrů pro skript *add-imscore-user_newdb.sh*:

`-u <user>` jméno uživatele např. "hanka",
`-t <tel:URI>` jednotný identifikátor zdroje např. "tel:+420777123456" - pokud tento parametr použijeme v kombinaci s parametrem *-a*, potom je vyžadován i parametr *-d*,
`-r <realm>` oblast – ve výchozím nastavení "open-ims.test" ,
`-i <impi>` soukromá identita např. "hanka@open-ims.test"- parametr *-u* tento přepíše,
`-b <impu>` veřejná identita např. "sip:hanka@open-ims.test"- parametr *-u* tento přepíše,
`-p <password>` heslo – ve výchozím nastavení údaj z parametru *-u*,
`-a` automatické použití námi vytvořeného skriptu " *add-user-hanka.sql* ",
`-d` automatické použití námi vytvořeného skriptu " *delete-user-hanka.sql* ",
`-c` odstranit skript po vytvoření uživatele (ve výchozím nastavení se neodstraňují).

Druhou možností vytvoření nového uživatele je pomocí webového rozhraní k HSS na adrese *http://localhost a portu 8080*. Následuje postup pro vytvoření nového uživatele "tom". V tomto případě už je zapotřebí mít spuštěné grafické rozhraní, pokud jsme tak neučinili již dříve. Jak bylo popsáno již dříve, do jednoduchého grafického režimu TWM se dostaneme zadáním příkazu *startx* nebo můžeme spustit přímo grafické rozhraní KDE, pokud jej naše distribuce obsahuje, a to pomocí příkazu */etc/init.d/xdm start*. Nyní spustíme libovolný webový prohlížeč (např. Mozilla Firefox) a zadáním adresy *http://localhost:8080* přistoupíme k ovládání databáze uživatelů HSS přes webové rozhraní.

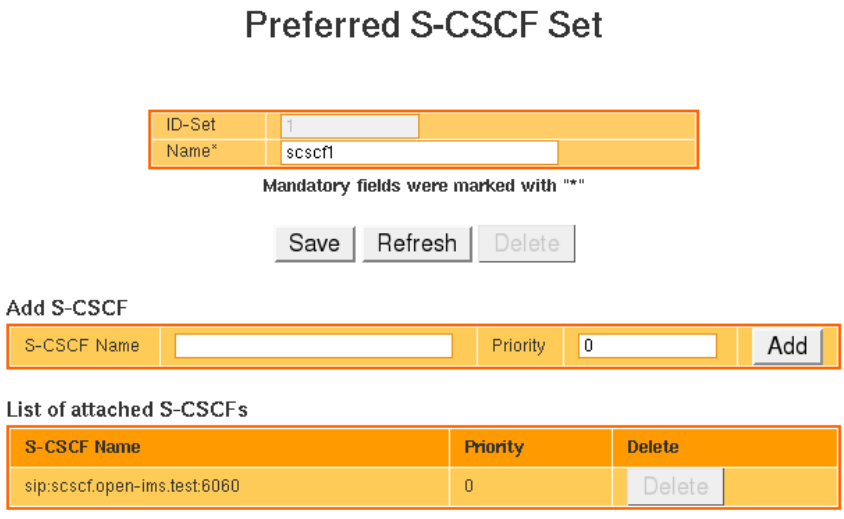
Nejprve se musíme přihlásit pomocí následujících přihlašovacích údajů - login: *hssAdmin*, password: *hss*. Po přihlášení můžeme v horním menu přepínat mezi záložkami Home - User Identities - Services - Network Configuration - Statistics.

Po vybrání záložky User Identities se nám otevře v lévé části okna nové menu s položkami IMS Subscription - Private Identity - Public User Identity. Z jednotlivých názvů je patrné, že se zde jedná o nastavení uživatelovy identity, jeho soukromé a veřejné identity. Pro vytvoření nového uživatele "Tom" nejprve klikneme na položku Create v menu IMS Subscription. Zde je nutné vyplnit pole *Name: tom*, tedy jméno uživatele. Dále položku *Capabilities Set*, jenž představuje sadu přiřazených vlastností, vybereme zde *cap_set1* a nakonec preferovaný server S-CSCF v položce *Preferred S-CSCF*, zde vybereme *scscf1*, jak je vidět na Obr. 5-8.



Obr. 5-8 Přidání IMSU tom

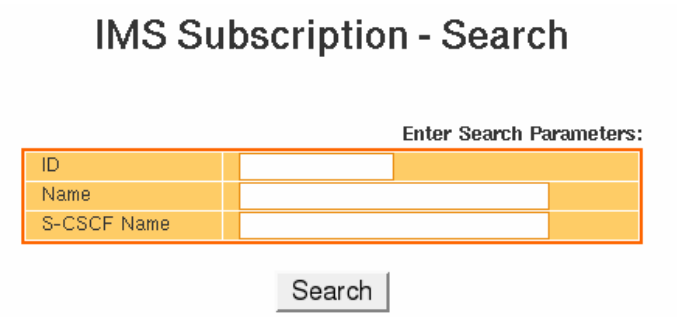
Jak vypadá nastavení vybrané položky *cap_set1* se můžeme podívat, když v hlavním menu vybereme záložku *Network Configuration* a zde vybereme z menu položku *Capabilities Set - Search - Name: cap_set1* a uvidíme kompletní výpis nastavení, jenž se skrývá pod položkou *cap_set1* (Obr. 5-10). Obdobně pro detailní pohled na vybranou položku *scscf1* vybereme z tohoto menu *Preferred S-CSCF Sets - Search - Name: scscf1* (Obr. 5-9).



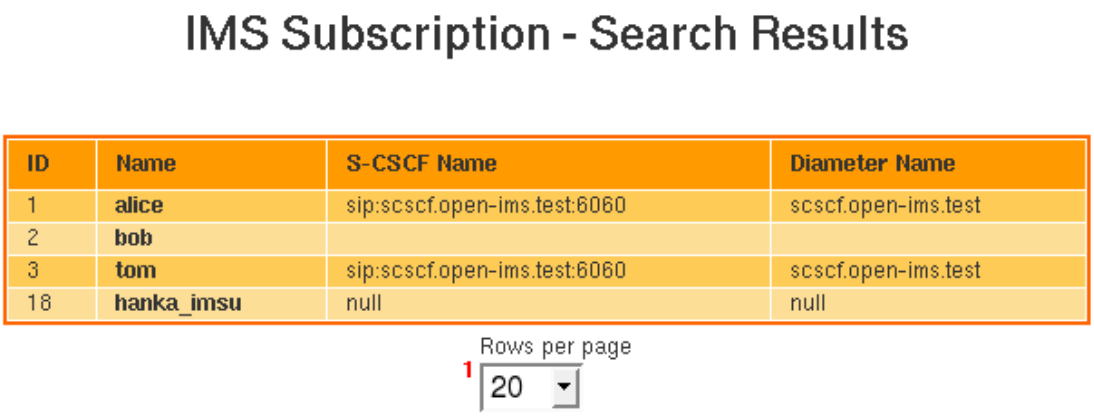
Obr. 5-9 Nastavení položky scscf1

Obr. 5-10 Nastavení položky cap_set1

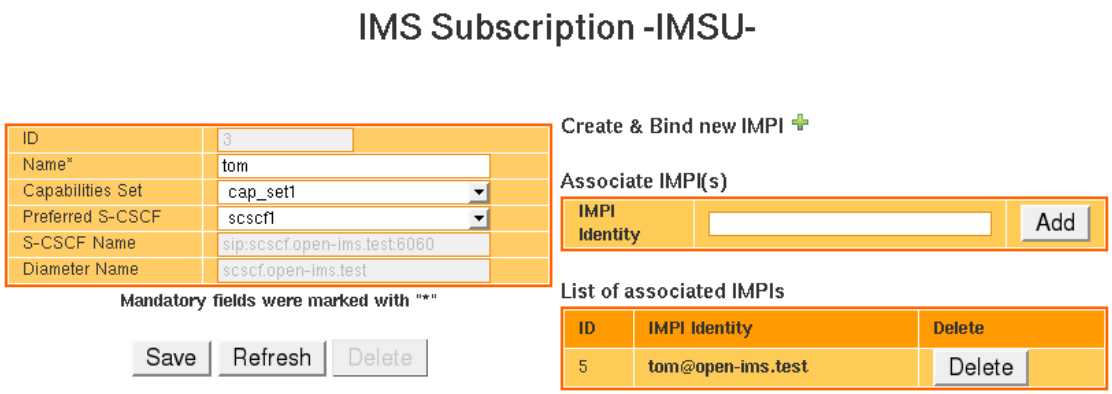
Námi vytvořené IMSU si můžeme ihned zobrazit, pokud v menu IMS Subscription klikneme na položku *Search*. Vyhledávání je umožněno hned několika způsoby. Můžeme zde vyhledávat pomocí ID uživatele, jména anebo pomocí serveru S-CSCF, jak vidíme na Obr. 5-11. Pokud si nejsme ani jednou položkou jisti a chceme si nechat vypsat všechny vytvořené uživatele, tak zvolíme volbu *Search* bez jakéhokoliv vyplňování. Zobrazí se nám kompletní výpis, kde navíc u již registrovaných uživatelů (v našem případě *alice* a *tom*) můžeme vidět také jméno S-CSCF serveru - *S-CSCF Name* a položku *Diameter Name* jako na Obr. 5-12. Následným výběrem našeho uživatele vidíme jeho nastavení, které zde můžeme i změnit, viz. Obr. 5-13.



Obr. 5-11 Vyhledávání IMSU

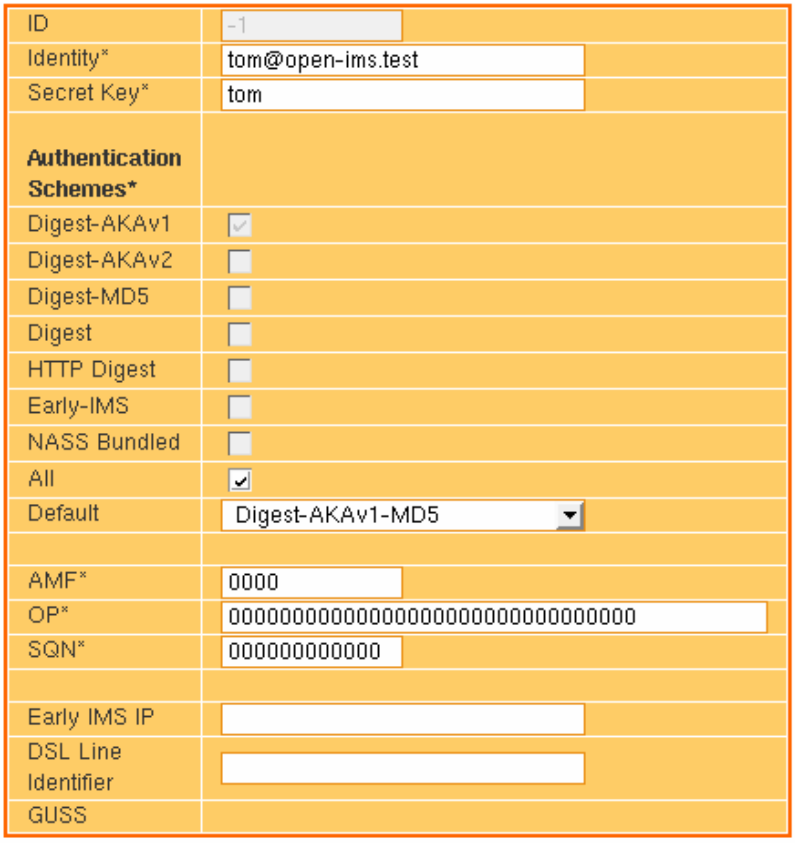


Obr. 5-12 Přehled všech dostupných IMSU

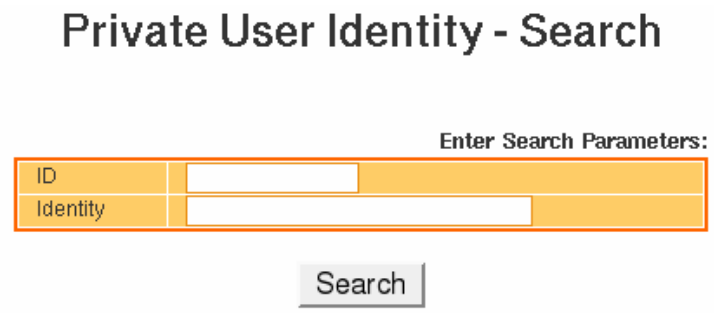


Obr. 5-13 Detail nastavení IMSU pro uživatele "tom"

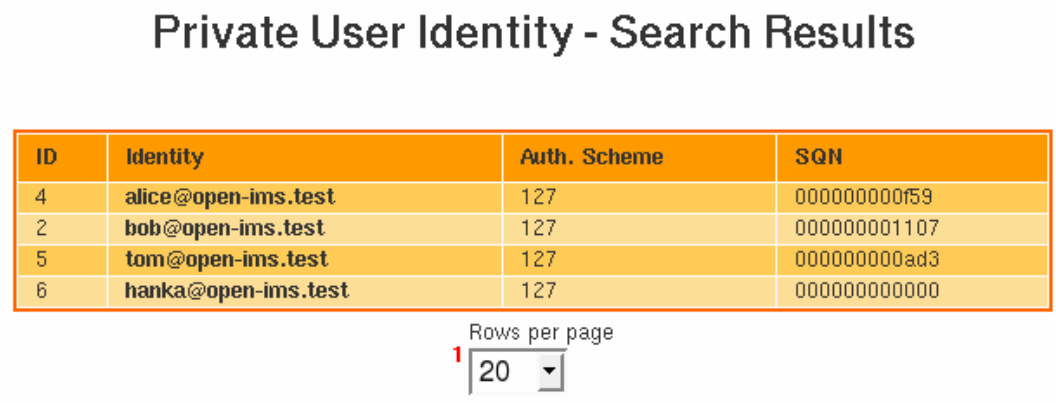
Nyní vytvoříme soukromou identitu uživatele, kterou přiřadíme uživateli tom. Vrátíme se opět do záložky *User Identities* a zde vybereme z menu *Private Identity* položku *Create*. Vyplníme zde pole *Identity: tom@open-ims.test*, tajný klíč *Secret Key: tom*, v položce Authentication Schemes si vybereme autentizační schéma, které chceme používat. Můžeme zde zvolit i více autentizačních schémat, přičemž v poli Default nastavíme to, které chceme používat ve výchozím stavu, u nás tedy zvolena výchozí hodnota Digest-AKAv1-MD5. Dále zde máme položky AMF - Authentication Management Field, jenž se využívá při autentizaci. V našem případě má délku 4B a nastavená hodnota *0000*. Položka OP - Operator ID, tedy ID operátora má 32B a má hodnotu *00000000000000000000000000000000*. Položka SQN - Sequence Number představuje sekvenční číslo o délce 12B s nastavenou hodnotou 000000000000. Veškeré nastavení vidíme na Obr. 5-14. Po vytvoření soukromé identity si ji opět můžeme zobrazit z menu *Private Identity - Search* kde můžeme vyhledávat buď podle ID nebo identity, viz. Obr. 5-15 a Obr. 5-16.



Obr. 5-14 Vytvoření IMPI pro uživatele "tom"

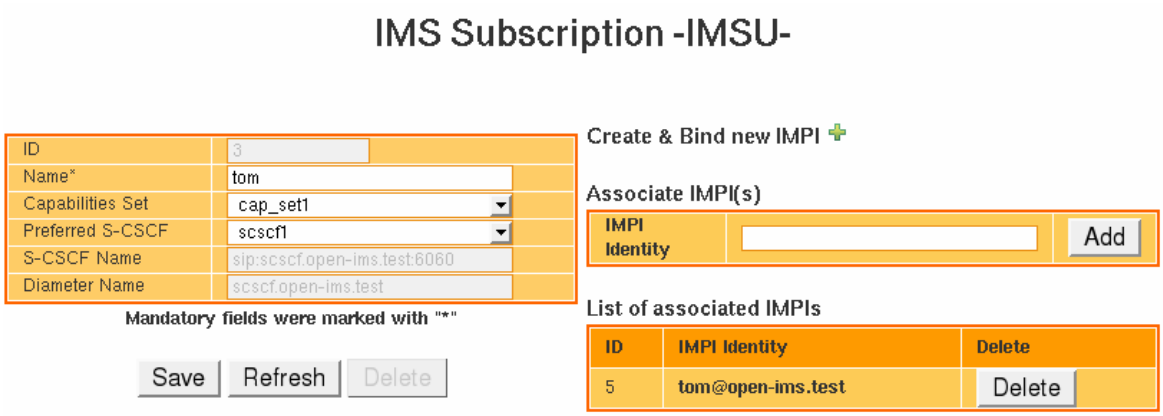


Obr. 5-15 Vyhledávání IMPI



Obr. 5-16 Přehled všech dostupných IMSU

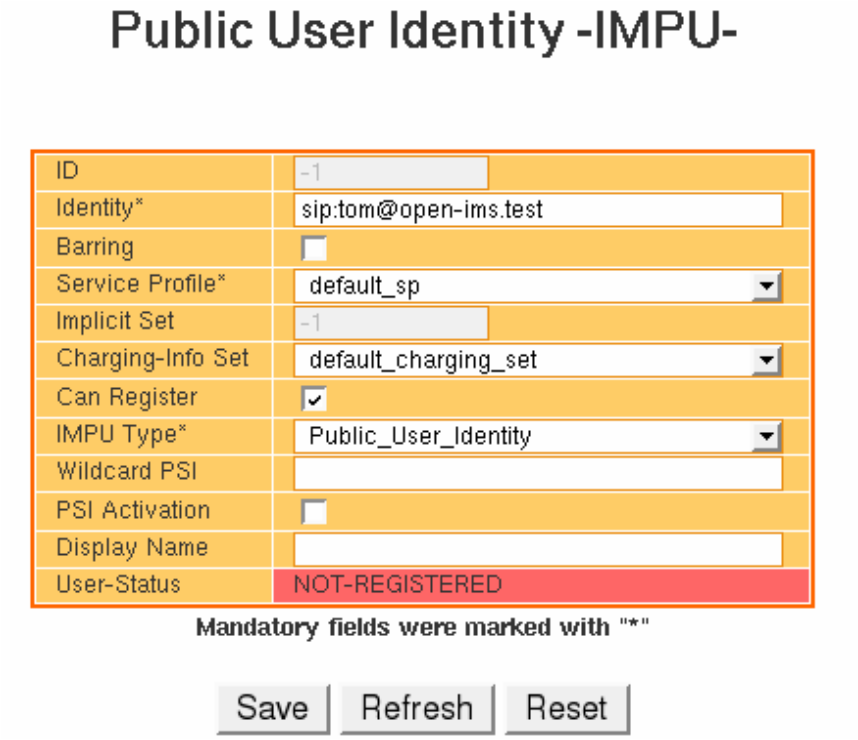
Přiřazenou soukromou identitu danému uživateli můžeme také vidět v menu *IMS Subscription*, kde si vyhledáme našeho uživatele. Zde pak nalezneme položku *List of associated IMPIs*, která představuje přehled soukromých identit pro daného uživatele. Je zde i možnost *Create & Bind new IMPI*, pomocí které můžeme vytvořit soukromou identitu daného uživatele stejně jako v předchozím kroku. Vše je názorně vidět na Obr. 5-17.



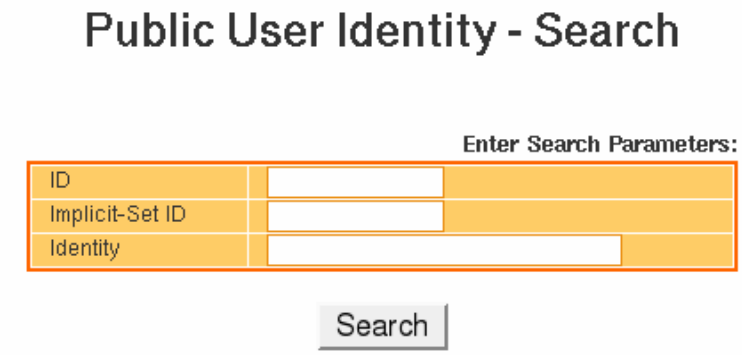
Obr. 5-17 Přiřazení IMPI danému uživateli

Posledním krokem je z menu *User Identities* položka *Public User Identity*, jenž slouží pro nastavení veřejné identity uživatele. Zde zvolíme položku *Create* pro vytvoření nové veřejné identity. Vyplníme pole *Identity: sip:tom@open-ims.test*. Dále políčko Barring necháme odškrtnuté, jedná se o doplňkovou službu pro omezení volání např. příchozích nebo odchozích hovorů. Položka *Service Profile* představuje profil služeb a nastavíme ji hodnotu *deafult_sp*, nastavení spojené s informacemi o účtování necháme rovněž ve výchozím stavu, takže pro položku *Charging-Info Set* vybereme *default_charging_set*. Zaškrtneme políčko *Can Register* pro možnost registrace. Pro typ veřejné identity možností *IMPU Type* vybereme *Public_User_Identity*, jelikož ta je určena pro IMS klienty. Položka *User-Status* nám poskytuje informaci o tom, zda je uživatel registrován či nikoliv. Veškeré nastavení je zobrazeno na Obr. 5-18.

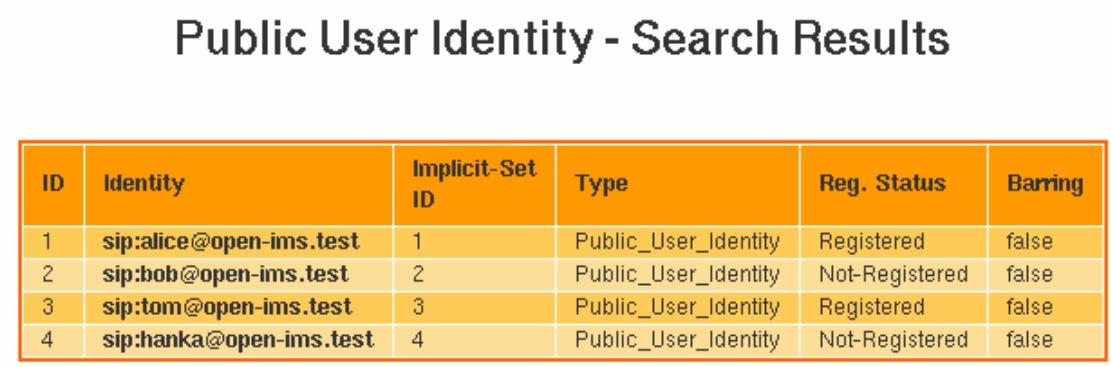
Po vytvoření veřejné identity si ji opět můžeme zobrazit z menu *Public Identity - Search* kde můžeme vyhledávat buď podle ID, implicitně nastaveného ID nebo podle identity, viz Obr. 5-19 a Obr. 5-20.



Obr. 5-18 Vytvoření IMPU pro uživatele "tom"

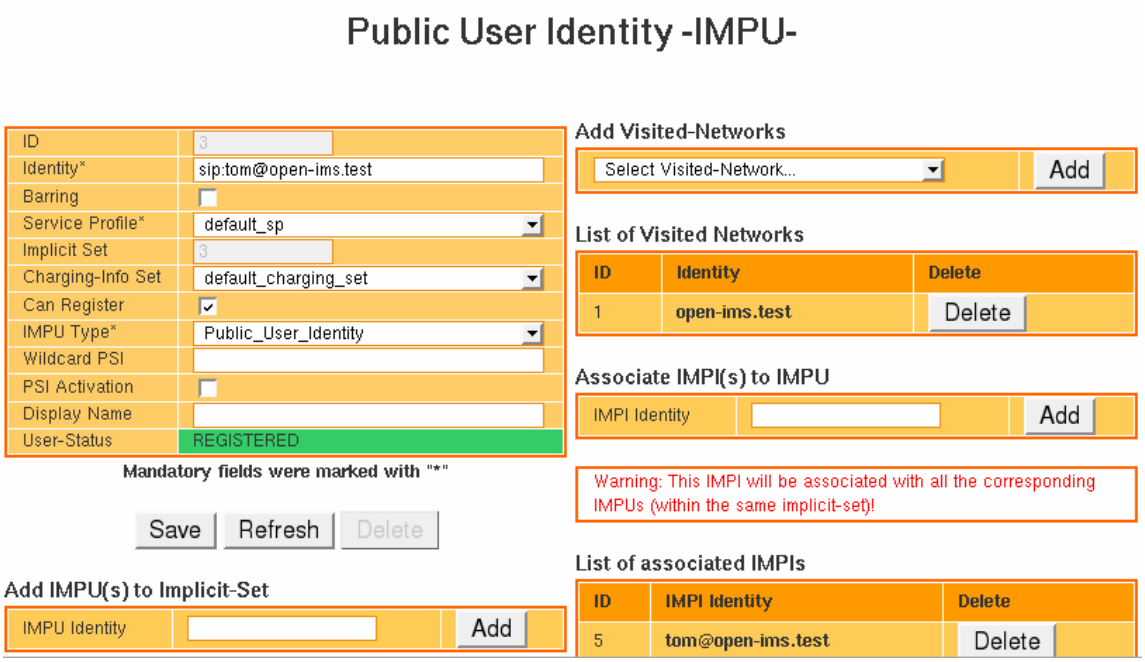


Obr. 5-19 Vyhledávání IMPU



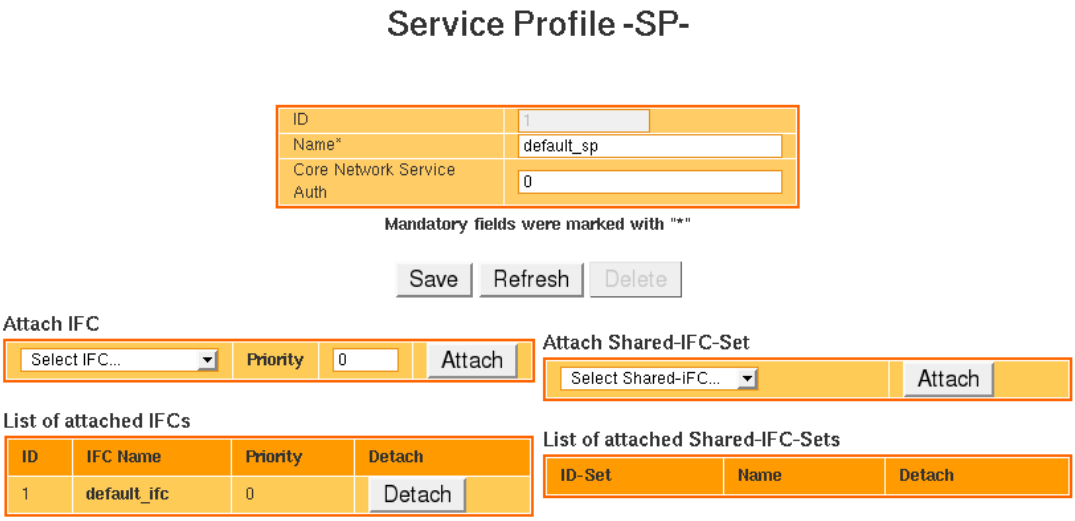
Obr. 5-20 Přehled všech dostupných IMPU

Jak vidíme v našem případě na Obr. 5-21 je uživatel tom s veřejnou identitou *sip:tom@open-ims.test* registrován. Napravo vidíme pod položkou *List of Visited Networks* seznam navštívených sítí - *open-ims.test*. Rovněž zde můžeme vidět seznam všech přidružených IMPI pod položkou *List od Associated IMPIs*, v našem případě zde máme soukromou identitu *tom@open-ims.test*.



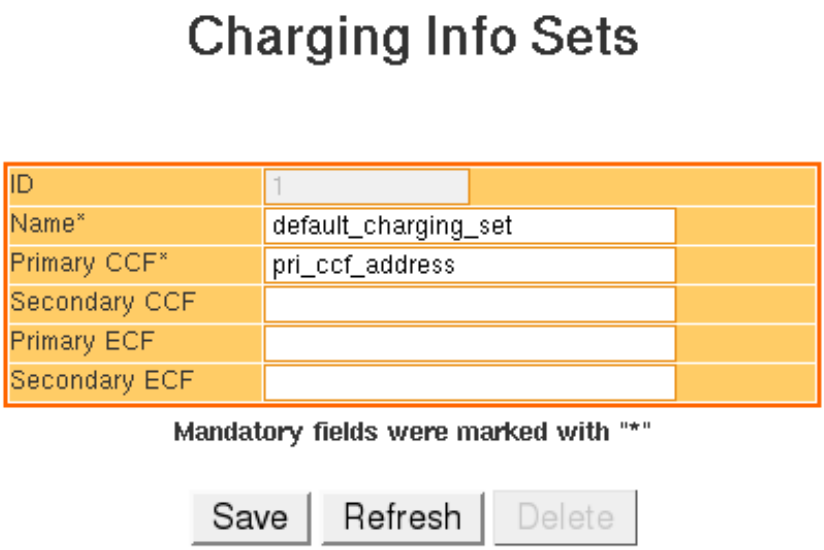
Obr. 5-21 Detail nastavení IMPU pro uživatele "tom"

Pokud se přepneme do záložky *Services* a z menu vybereme *Service profiles* - *Name* - *default_sp*, tak se můžeme podívat na detail nastavení profilu služeb, viz. Obr. 5-22.



Obr. 5-22 Nastavení profilu služeb

V záložce *Network Configuration* si můžeme z menu *Charging Sets* nechat vypsat nastavení spojené s účtováním. Zvolíme *Search - Name - default_charging_set*. Konkrétní nastavení vidíme na Obr. 5-23.



Obr. 5-23 Nastavení spojené s účtováním

5.4 Komunikace v IMS síti

V předchozí kapitole jsme si ukázali, jak lze v systému Open IMS Core vytvořit nové uživatele. Nyní, když máme tyto uživatelské účty vytvořeny, můžeme přistoupit k vlastní komunikaci mezi těmito uživateli. K tomuto bude potřeba nainstalovat IMS klienty. Existuje zde několik IMS klientů, kteří můžou být použiti ve spojení se systémem Open IMS Core, např. *IMS Communicator*, *UCT IMS Client* a nebo *Open IMS Client*. Poslední zmiňovaný je dílem Fraunhoferova Institutu FOKUS, jenž vyvinul i systém Open IMS Core. Open IMS Client je dostupný jen v komerční verzi, ale pro naše potřeby testování plně poslouží i jeho volná verze *OpenIC_Lite*.

Nejdříve si tedy stáhneme a nainstalujeme IMS klienta OpenIC_Lite. Instalační balíček se jmenuje *OpenIC_Lite.tar.gz*. Po uložení přejdeme do daného adresáře a rozbalíme buď pomocí Midnight Commanderu (utilita *mc*) nebo pomocí následujícího příkazu:

[13]

```
tar –zxvf OpenIC_Lite.tar.gz
```

Po rozbalení nesmíme ještě zapomenout na přidělení práv ke spouštěcímu souboru OpenIC_Lite.sh. Opět můžeme toto nastavení provézt buď pomocí utility *mc* nebo pomocí následujícího příkazu:

```
cd /opt/openIMSCore/OpenIC_Lite/
chmod 744 OpenIC_Lite.sh
```

Tímto zajistíme práva pro vlastníka RWX, pro skupinu R-- a pro ostatní R--. Následující Tab 5.1 nám význam tohoto příkazu vysvětlí blíže.

Tab 5.1 Přístupové práva

User	Group	Others
4 2 1 RWX	4 2 1 RWX	4 2 1 RWX
R = Read - čtení / W = Write -zápis / X = Execute - spuštění		

Nyní již máme přidělena všechna potřebná práva a můžeme tedy spustit spouštěcí skript opět pomocí utility *mc* nebo pomocí následujícího příkazu:

```
./ OpenIC_Lite.sh
```

Při spuštění můžeme ještě narazit na problém týkající se nastavení proměnné JAVA_HOME ve spouštěcím skriptu *OpenIC_Lite.sh*. Tato chyba může vypadat následovně:

```
OpenIMS ~ # mc
OpenIMS OpenIC_Lite # ./OpenIC_Lite.sh
./OpenIC_Lite.sh: line 20: /root/bin/java: No such
file or directory
```

Je zde nutné zadat této proměnné cestu, kde se v našem systému nachází systémové prostředí JAVA. Pokud bychom tak neučinili, tak spouštěcí skript nebude schopen otevřít klienta, jenž ke svému spuštění využívá právě nástroje JAVA. V našem případě bylo třeba nastavit proměnné JAVA_HOME následující cestu:

```
JAVA_HOME: /usr/lib/jvm/sun-jdk-1.5/
```

Před samotným spuštěním klienta musíme mít samozřejmě nastartovány všechny potřebné CSCF servery a databázový server HSS, pokud by jsme tak neučinili, klient by se nebyl schopen zaregistrovat do IMS sítě. V takovém případě provedeme následující kroky:

- spustíme jednotlivé servery CSCF, viz. Obr. 5-24, Obr. 5-25 a Obr. 5-26. ,

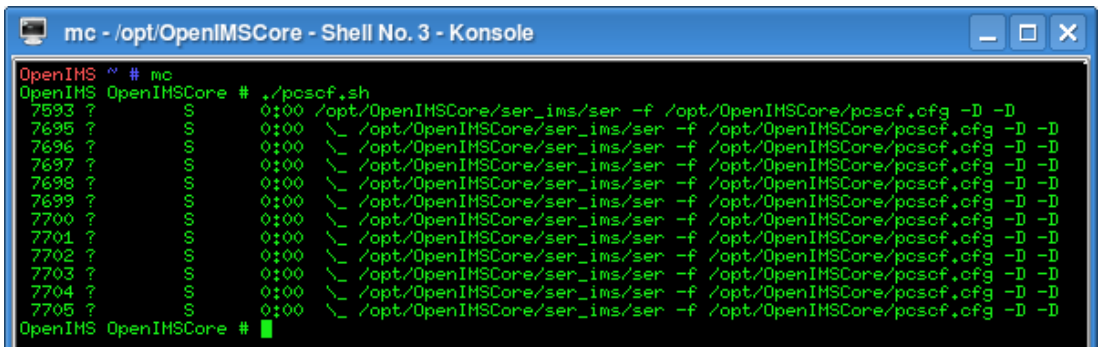
```
cd /opt/OpenIMSCore
./pcscf.sh
./icscf.sh
./scscf.sh
```

- spustíme server s databází uživatelů, viz. Obr. 5-27.,

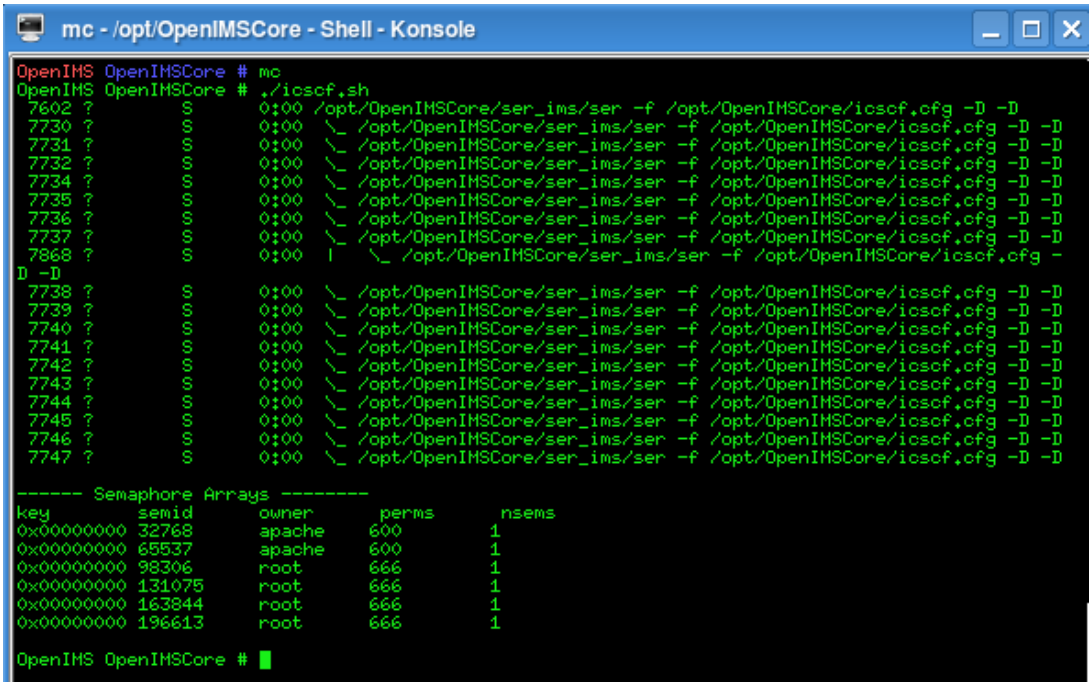
```
cd /opt/OpenIMSCore/FHoSS/deploy
./startup.sh
```

- a poté samotného IMS klienta.

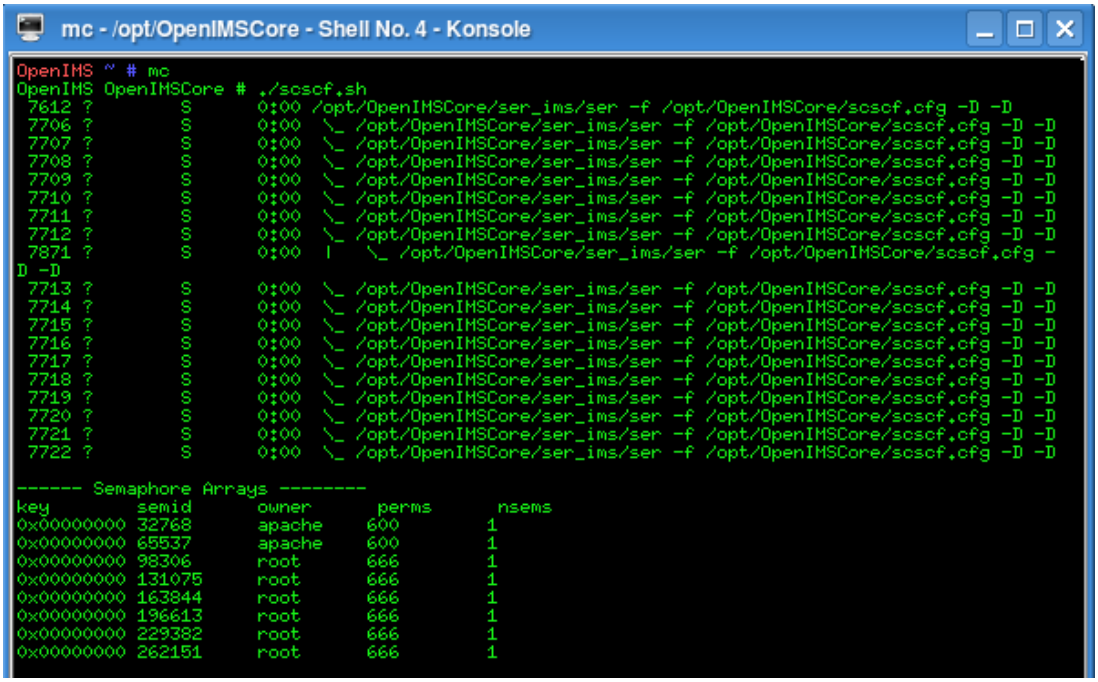
```
cd /opt/openIMSCore/OpenIC_Lite/
./ OpenIC_Lite.sh
```



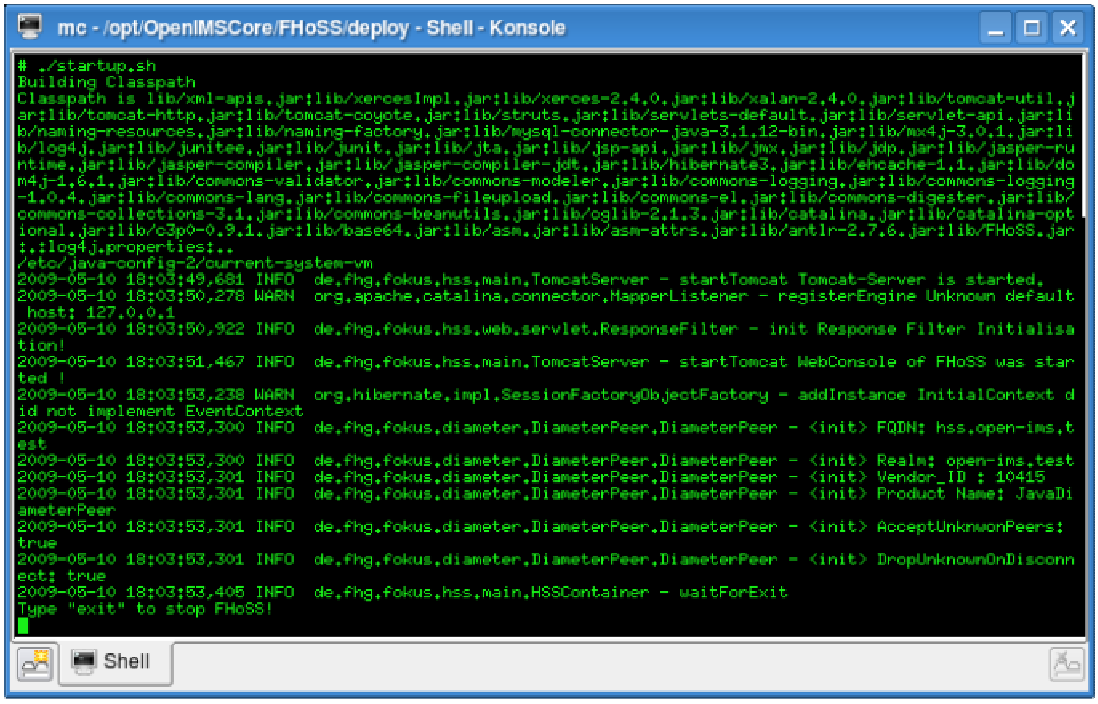
Obr. 5-24 Spuštění proxy serveru P-CSCF



Obr. 5-25 Spuštění serveru I-CSCF



Obr. 5-26 Spuštění serveru S-CSCF



Obr. 5-27 Spuštění serveru HSS

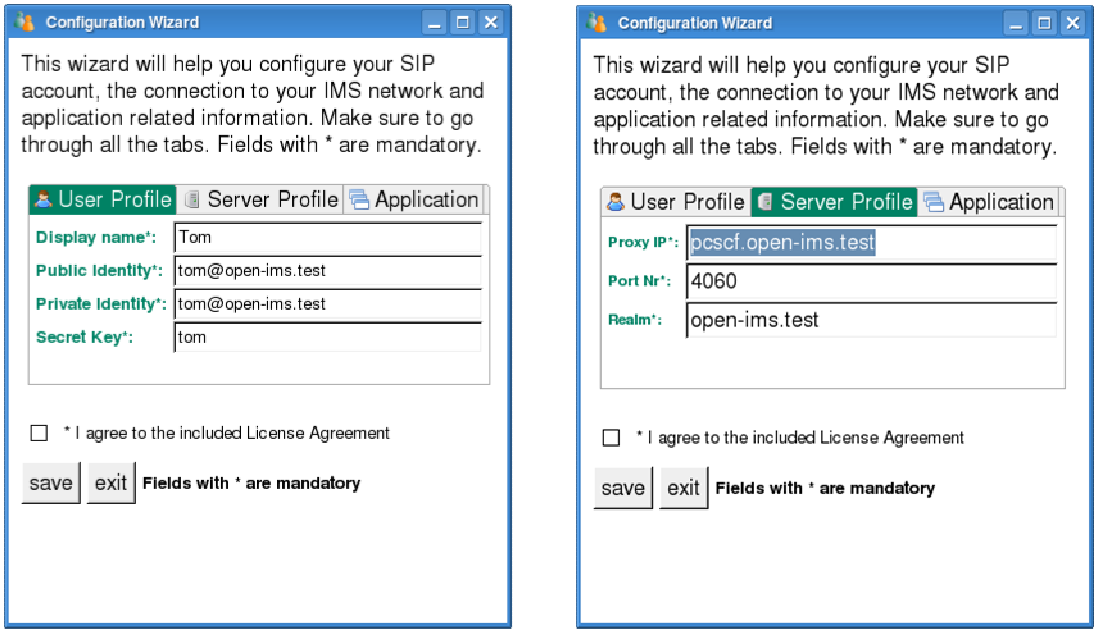
Při spouštění serveru HSS se může vyskytnout chyba, hlásící obsazený port 8080. Tato chyba může vypadat následovně:

```
Java.net.BindException: Address already in use:8080
```

K uvolnění portu 8080 musíme znát službu, která tento port obsadila. K tomuto nám poslouží příkaz *netstat* s příslušnými parametry, který nám vypíše název služby na daném portu spolu s jeho PID (Process ID). Poté příkazem *kill* tuto službu ukončíme.

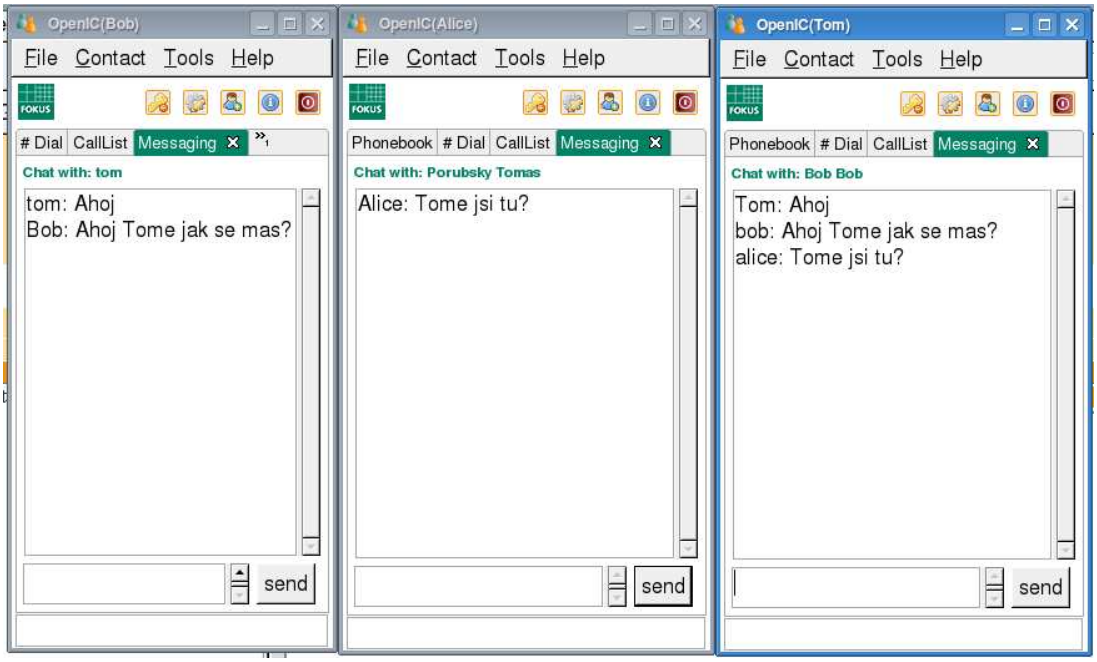
```
netstat -anp | grep 8080
kill <PID>
```

Po spuštění klienta nám najede do konfiguračního okna pro konfiguraci uživatelského účtu. Jak vidíme na Obr. 5-28, toto okno obsahuje tři záložky *User Profile*, *Server Profile* a *Application*. V *User Profile* definujeme zobrazované jméno pro uživatele (Tom), dále pak jeho veřejnou identitu (tom@open-ims.test), soukromou identitu (tom@open-ims.test) a tajný klíč (tom). V záložce *Server Profile* definujeme adresu proxy serveru (pcscf.open-ims.test), číslo portu (4060) a oblast pro rozřazení účastníků (open-ims.test). V poslední záložce *Application* máme možnost nastavit automatické přijímání hovorů, automatickou registraci klienta a možnost zapnout výstražná hlášení. Kompletní nastavení je pak nutné po souhlasu s licencí uložit tlačítkem *save*. Tímto se nám vytvoří konfigurační soubor *profile.cfg* v adresáři */opt/OpenIMSCore/OpenIC_Lite/*. Výpis souboru můžeme vidět v příloze A.



Obr. 5-28 Konfigurační okno IMS klienta

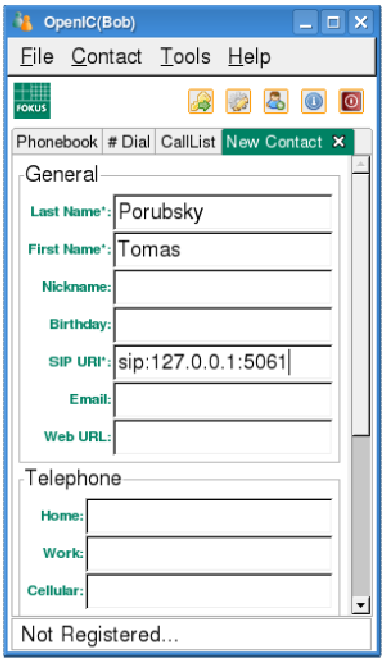
Pokud jsme v předchozím kroku nezaškrtli políčko s automatickou registrací, tak přejdeme do menu *File - Sign In*. Ve spodní části by se nám měl objevit text *Registered*, což nám oznamuje, že registrace tohoto účtu proběhla v pořádku. Toto si můžeme zkontrolovat i ve webovém rozhraní k HSS na již známé adrese *http://localhost:8080*. Zde vidíme v menu *User Identities - IMS Subscription - Search*, že zaregistrovaným účtům je přiřazen S-CSCF name a Diameter Name. Pod položkou *Public User Identity - Search* pak máme u našeho účtu *Reg. Status = Registered*, čili registrován.



Obr. 5-29 Ukázka komunikace mezi klienty

Na

Obr. 5-29 vidíme komunikaci mezi uživateli Bob - Alice - Tom. Nejjednodušší volbou je přidání si všech známých uživatelů do seznamu (*Phonebook*) v menu *Contact - Add Contact.* Zadáme povinné položky *Last Name*, *First Name* a *SIP URI* a volitelně můžeme email, telefon či adresu daného uživatele viz. Obr. 5-30. Poté již máme přístupné menu přes pravé tlačítko k zaslání zprávy (*Send Message*) či zahájení hovoru (*Call*).



Obr. 5-30 Přidání nového kontaktu do seznamu

Konzolový režim

OpenIC_Lite klienta máme možnost spustit i v konzolovém režimu. Pro tento režim je však nutné vytvořit konfigurační soubor pro daného uživatele IMS sítě ručně. Pro uživatele Tom je tento konfigurační soubor uložen v */opt/OpenIMSCore/OpenIC_Lite/Tom/profile.cfg*
a jeho výpis vypadá následovně:

```
displayName=Tom
publicIdentities=sip:tom@open-ims.test
realm=open-ims.test
privateIdentity=tom@open-ims.test
secretKey=tom
proxyCSCF=pcscf.open.ims-test:4060/UDP
SQN=000000000000
AMF=0000
AMFSTAR=0000
OP=00000000000000000000000000000000
useAK=true
simulateISIM=false
sqnVectorCurrentIndex=0
IND_LEN=5
delta=268435456
L=32
```

Jak vidíme, zadávají se zde parametry jako zobrazované jméno (*displayName*), veřejná identita (*publicIdentities*), oblast pro možné rozřazení účastníků (*realm*), soukromá identita (*privateIdentity*), tajný klíč (*secretKey*), proxy server (*proxyCSCF*), sekvenční číslo (*SQN*), *Authentication Management Field* využívající se při autentizaci (*AMF*) i (*AMFSTAR*), identifikátor operátora (*OP*), využití autentizačního klíče (*useAK*), simulace ISIM (*simulateISIM*), aktuální index sekvenčního vektoru (*sqnVectorCurrentIndex*) a parametry pro generování SQN (*IND_LEN, delta, L*).

Přehled možných příkazů v konzolovém režimu:

Tab 5.2 Přehled příkazů v konzolovém režimu

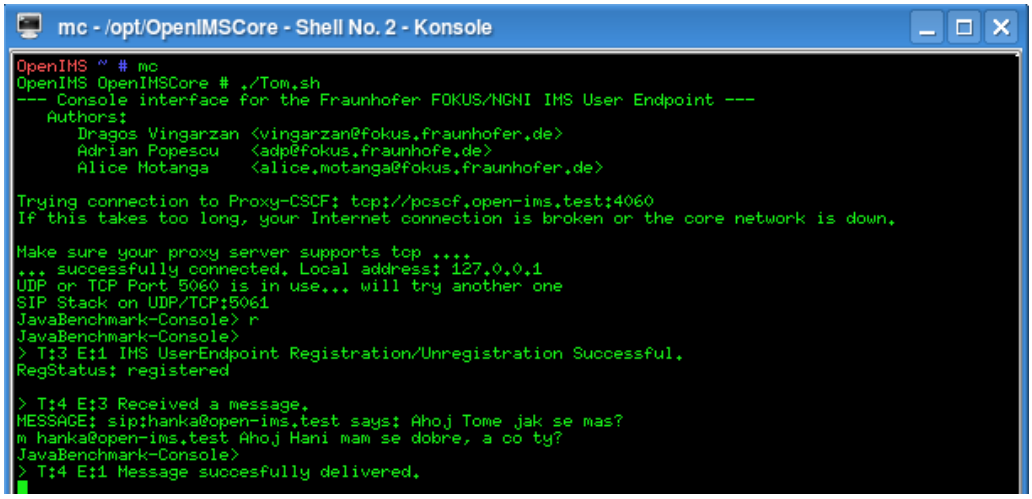
x	Ukončí aplikaci (alias: x, exit).
r <expires>	Registrace UE (alias: r, reg, register).
ur	Odregistrování UE (alias: ur, ureg, unreg, unregister).
ura	Odregistrování všech kontaktů (alias: ura, urega, unregall, unregisterall).
p <stat>[<exp>]	Zveřejnit informace o sobě (alias: p, pub, publish).
up	Nezveřejnit informace o sobě (alias: up, upub, unpub, unpublish).
s <uri>[<exp>]	Podepsat se pod informace od <uri> (alias: s, subs, subscribe).
us <uri>	Nepodepsat se pod informace od <uri> (alias: us,usubs, unsubs, unsubscribe).
usa	Zrušit veškeré předchozí podepsání (alias: usa, usubsa, unsubsa, unsubscribeall).
sr[<exp>]	Podepsat [reg] události na serveru S-CSCF (alias: sr, subsreg, subscribereg).
m <uri> <msg>	Zaslat <uri> zprávu <msg> (alias: m, msg, message).
c <uri>	Volat <uri> (alias: c, call).
y <id>	Přijmout hovor s <id> (alias: y, yes, answer).
n <id>	Odmítnout hovor s <id> (alias: n, no, reject).
a <id>	Přerušit hovor s <id> (během vyzvánění) (alias: a, abort).
h <id>	Ukončit hovor s <id> (alias: h, hang, hangup).
l	Tabulka seznamu hovorů (alias: l, list).
ls	Seznam předplatného se stavem (alias: ls, listsubs).
lr	List current Reginfo (alias: lr, listreg).
lt	Seznam časovačů UE (alias: lt, listtimers).

Pro spuštění v konzolovém režimu vytvoříme spouštěcí skript se jménem daného uživatele v adresáři /opt/OpenIMSCore. Pro uživatele "Tom" je skript uveden v příloze A.

Nyní máme vše připraveno pro spuštění klienta v konzolovém režimu. Přejdeme tedy do adresáře */opt/OpenIMSCore/* a spustíme námi vytvořený skript *Tom.sh*.

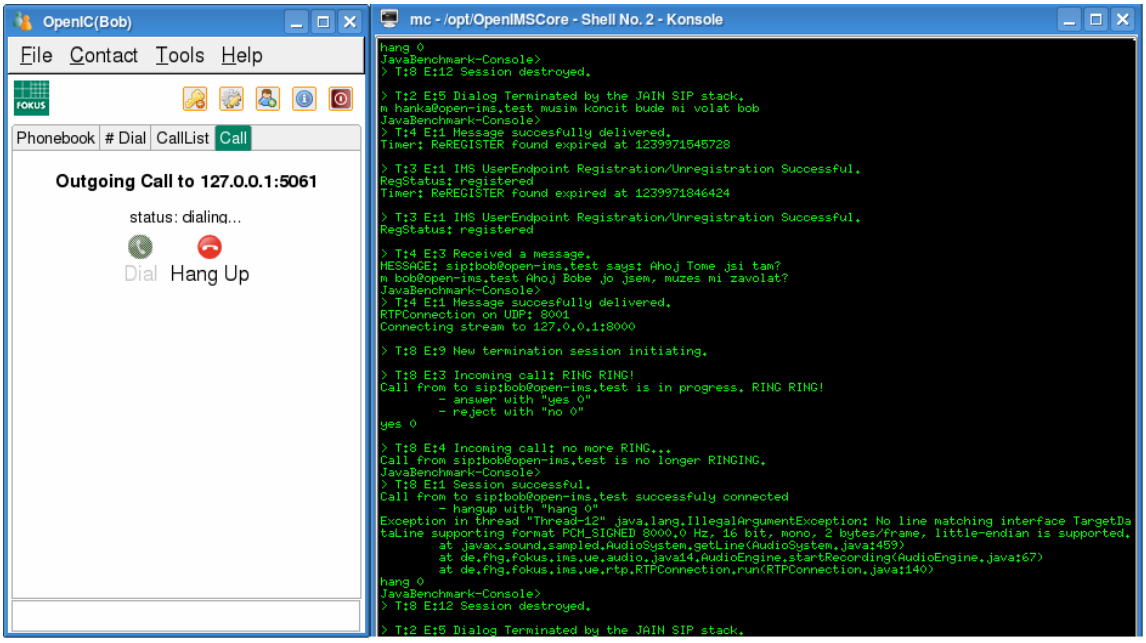
```
cd /opt/OpetIMSCore
./ Tom.sh
```

Na Obr. 5-31 vidíme spuštění klienta "Tom" v konzolovém režimu, jeho úspěšné připojení k proxy serveru P-CSCF a ustálení spojení TCP/UDP na portu 5061. Dále je zde jeho úspěšná registrace (`r`) a následná komunikace s uživatelem "Hanka" (`m <uri> <msg>`).



Obr. 5-31 Uživatel "Tom" v konzolovém režimu

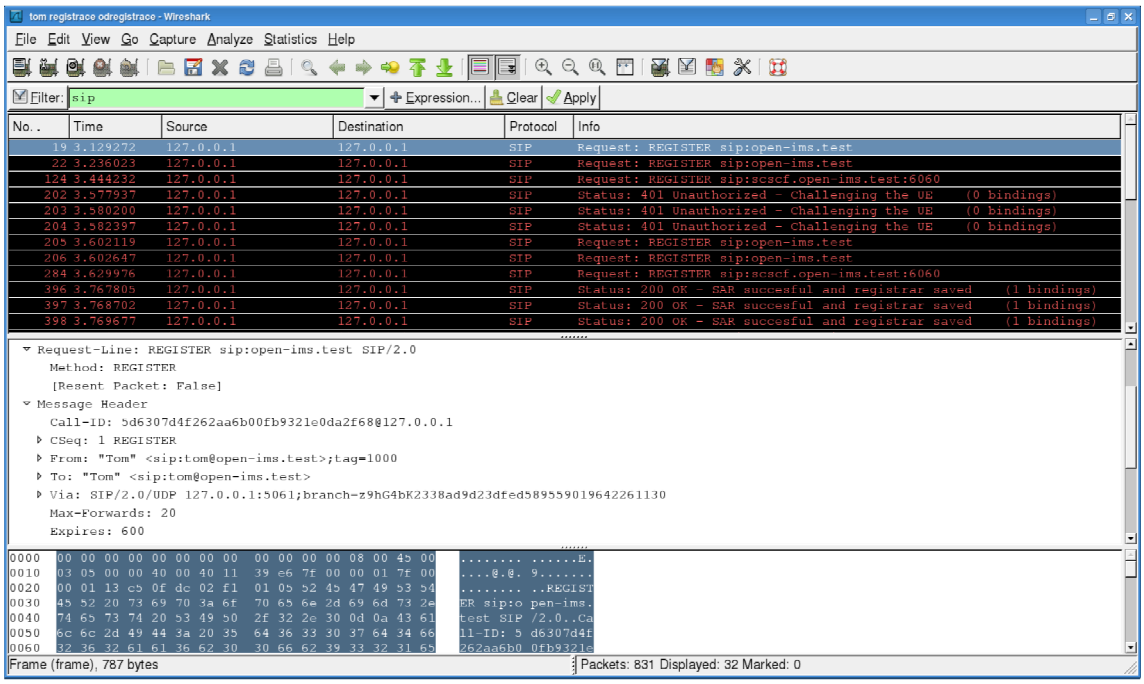
Na Obr. 5-32 vidíme jak uživatel Bob (klient v grafickém režimu) volá uživatele Toma (v konzolovém režimu) - *Outgoing Call to 127.0.0.1:5061*. Ve výpisu z konzoly vidíme vytvořené spojení na UDP portu 8001 a přicházející hovor s vyzváněním (*incoming Call: RING RING!)*. Tom přijme hovor příkazem *yes 0* a zavěsit hovor lze příkazem *hang 0*.



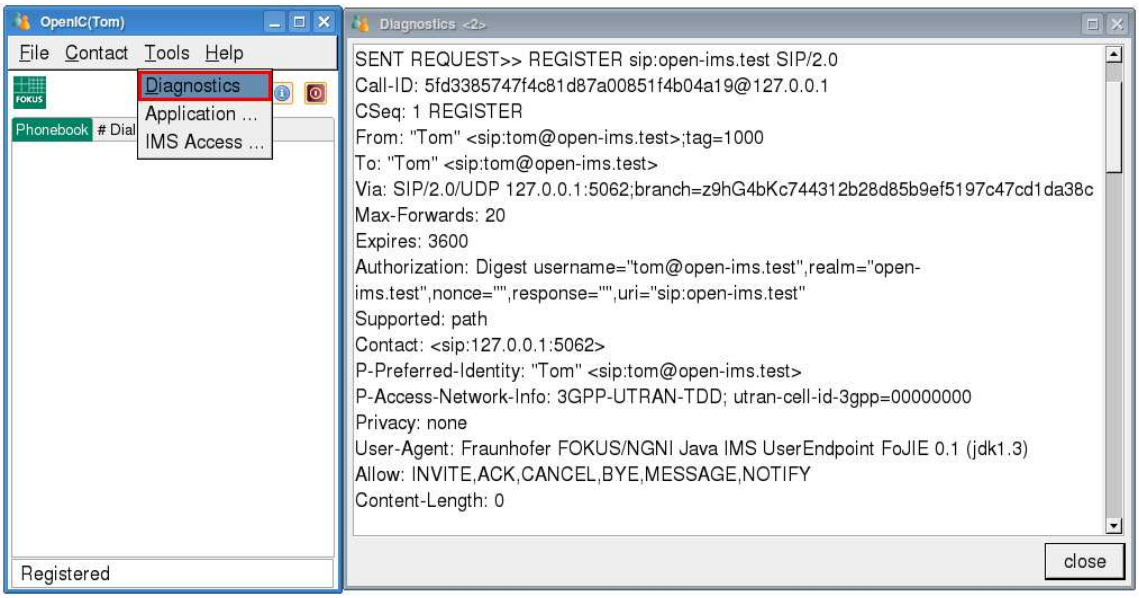
Obr. 5-32 Bob volá Toma, který tento hovor přijme

6 Analýza přenášených zpráv

Pro analýzu přenášených zpráv jsem si vybral aplikaci Wireshark, jenž v sobě ukrývá paketový zachytávač, tzv. sniffer a protokolový analyzátor v jednom. Jedná se o přímého nástupce aplikace Ethereal, jenž poskytuje vylepšenou analýzu VoIP komunikace, velké množství podporovaných protokolů a jejich následné dešifrování. Zachycení registrace ve Wiresharku můžeme vidět na Obr. 6-1. Kromě Wiresharku lze pro analýzu komunikace v IMS síti použít také nástroje obsažené v samotném klientovi OpenIC_Lite, přístupné v menu *Tools - Diagnostics*. Zde se však jedná o zachytávání komunikace jen mezi IMS terminálem a P-CSCF, viz. Obr. 6-2. Také webové rozhraní HSS poskytuje jednoduchou analýzu přenášených zpráv a to právě mezi HSS a servery I-CSCF a S-CSCF. Zachytávání a následnou analýzu spustíme z menu *Statistic - Turn On Debug*, jak vidíme na Obr. 6-3.

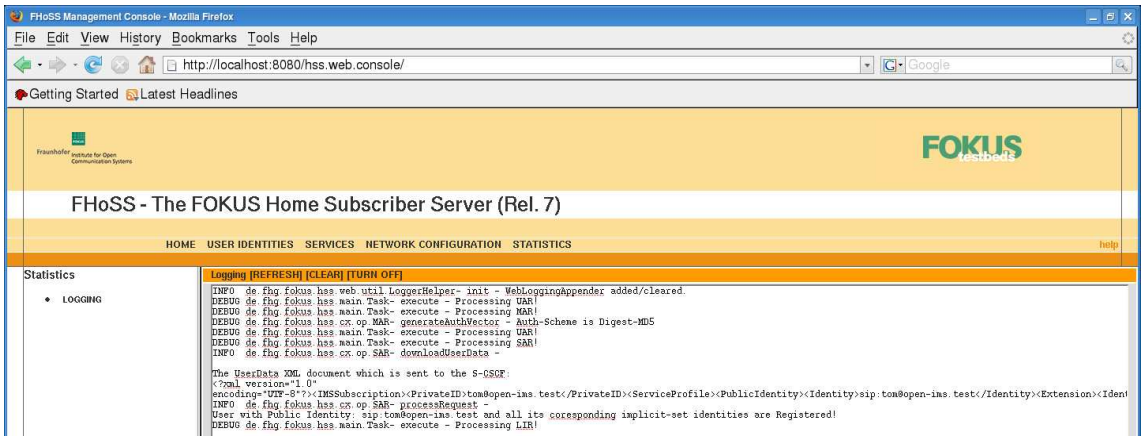


Obr. 6-1 Zachycení registrace ve Wiresharku



Obr. 6-2 Zachycení registrace v OpenIC_Lite

Kompletní výpis zachycených zpráv při registraci uživatele Tom pomocí klienta OpenIC_Lite najdeme v příloze C.



Obr. 6-3 Zachycení registrace ve webovém rozhraní HSS

6.1 Počáteční registrace v IMS síti

Teoretický průběh počáteční registrace uživatele v IMS síti spolu s popisem průběhu jsem už popsal dříve v kapitole 4.1.2, viz. Obr. 4-2. Zde si ukážeme, jak vypadá registrace v našem konkrétním případě pro uživatele "Tom", který je na portu 5061. V následujících výpisech průběhu registrace jsou uvedeny i čísla portů, jenž jsou daným serverům a službám přiřazeny následovně:

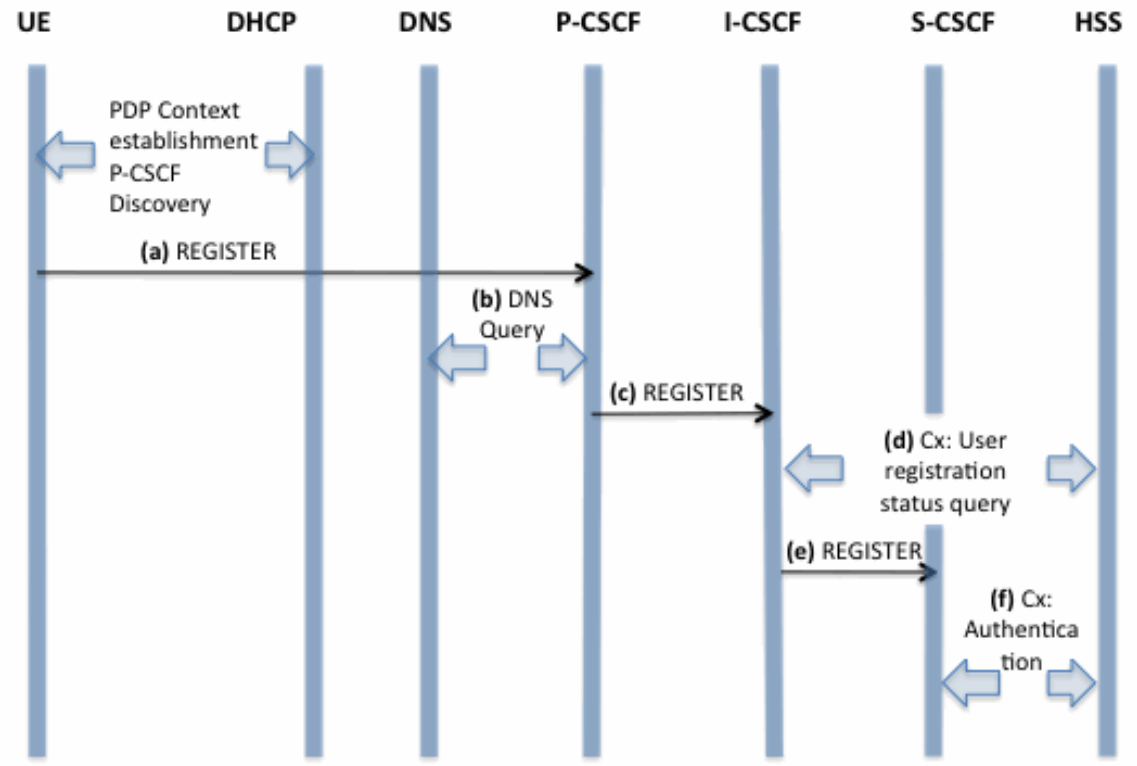
[12, 14]

Tab 6.1 Čísla portů jednotlivých serverů a služeb

číslo portu	server/služba
53	DNS
4060	P-CSCF
5060	I-CSCF
5061	UE klient
6060	S-CSCF
32770	TCP/UDP

Registrace uživatele "Tom":

I. Zpráva REGISTER



Obr. 6-4 Zaslání zprávy REGISTER

Ještě před samotnou registrací uživatele, musí UE získat IP spojení a nalézt vstupní bod do IMS, tedy P-CSCF. Níže vidíme tuto komunikaci zachycenou v programu Wireshark. Je zde uveden čas odeslání paketu- *Time* a rozhraní, ze kterého byl paket odeslán - *127.0.0.1* spolu s číslem portu. Zde konkrétně výměna DNS informací o serveru P-CSCF na rozhraní 127.0.0.1, portech 32770 a 53 v časech 0.000 a 0.001.

```
|Time     | 127.0.0.1          |
|0.000    |             Standard query AAAADNS: Standard query
                        AAAA pcscf.open-ims.test
|         |(32770)-------------------------------------> (53)

|0.000    |             Standard query respDNS: Standard query
                        response
|         |(53)   -------------------------------------->(32770)

|0.001    |             Standard query AAAADNS: Standard query
                        AAAA pcscf.open-ims.test
|         |(32770)-------------------------------------> (53)

|0.001    |             Standard query respDNS: Standard query
                        response
|         |(53)   -------------------------------------> (32770)

|0.001    |             Standard query A pcDNS: Standard query A
                        pcscf.open-ims.test
|         |(32770)-------------------------------------> (53)

|0.001    |             Standard query respDNS: Standard query
                        response A 127.0.0.1
|         |(53)   -------------------------------------> (32770)
```

a) zpráva REGISTER z UE do P-CSCF

Účelem této žádosti je registrovat SIP URI s S-CSCF v domácí síti. Tato žádost je směrována do serveru P-CSCF. Jak vidíme níže, komunikace probíhá na portu 5061, jenž představuje port našeho klienta a na portu 4060, na kterém najdeme server P-CSCF.

```
|Time     | 127.0.0.1
|3.129    |             Request: REGISTER sSIP: Request: REGISTER
                        sip:open-ims.test
|         |(5061) -------------------------------------> (4060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
REGISTER sip:open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5061;
branch=z9hG4bK2338ad9d23dfed589559019642261130
Max-Forwards: 20
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=00000000
From: "Tom" <sip:tom@open-ims.test>;tag=1000
To: "Tom" <sip:tom@open-ims.test>
Contact: <sip:127.0.0.1:5061>
Call-ID: 5d6307d4f262aa6b00fb9321e0da2f68@127.0.0.1
Authorization: Digest username="tom@open-ims.test", realm="open-
ims.test", nonce="", response="", uri="sip:open-ims.test"
CSeq: 1 REGISTER
Supported: path
Content-Length: 0
```

Request-URI - následuje za názvem metody ("REGISTER") v prvním řádku, značí cílovou doménu této žádosti REGISTER. Pravidla pro směrování SIP žádostí popisují, jak využít DNS k rozpoznání adresy nebo vstupního bodu domovské sítě operátora (I-CSCF) z tohoto doménového jména. Tato informace je uložena v USIM.

Via - adresa UE přidělená během procesu aktivace PDP kontextu.

P-Access-Network - UE poskytuje informace o přístupové metodě vztažené k obsluhující síti.

From - značí veřejnou identitu uživatele, který vyslal žádost REGISTER. Veřejná identita může být získána z USIM.

To - značí veřejnou identitu uživatele, který se registruje. Je to identita, podle které ostatní strany poznají tohoto účastníka.

Contact - označuje účastníka, IP adresu UE. Jedná se o dočasné označení pro účastníka, který se registruje. Následné žádosti pro tohoto účastníka budou zaslány na tuto adresu. Tato informace je uložena v S-CSCF.

Authorization - nese autentizační informace. Soukromá identita uživatele (tom@open-ims.test) je uložena v Digest AKA protokolu v poli *username*. Parmetr *uri* obsahuje stejnou hodnotu jako *Request-URI*. *Realm* obsahuje název sítě, ve které je účastník autentizován. Parametry *Request-URI* a *realm* jsou získány ze stejného pole z USIM, a proto jsou také identické. Parametry *nonce* a *response* jsou zde prázdné, což simuluje vložení nové UICC karty do terminálu, kdy nejsou k dispozici žádné jiné uložené informace k odeslání.

Supported - tohle záhlaví je zahrnuto v upozornění příjemce, že UE podporuje hlavičku Path.

b) DNS dotazy mezi P-CSCF a DNS: "DNS Query"

P-CSCF na základě URI uživatele zjistí, že UE se registruje z návštěvnické domény a provede DNS dotaz k určení I-CSCF v domácí síti. Během přeposílání žádosti REGISTER potřebuje server P-CSCF specifikovat protokol, číslo portu a IP adresu serveru I-CSCF v domácí síti, kterému tuto žádost zasílá. P-CSCF provede NAPTR (Name Autority PoinTeR) dotaz na doménu specifikovanou v *Request-URI*. Na základě pořadí a preferencí záznamu NAPTR se upřednostní UDP a P-CSCF nalezne I-CSCF pomocí vyhledání DNS SRV (Service). Právě SRV záznamy jsou totiž často použity k nalezení serveru, který poskytuje požadovanou službu.

V odpovědi na výzvu je každý I-CSCF server identifikován podle jeho doménového jména. Navrácené hodnoty záznamů SRV jsou sloučeny a seřazeny. Pomocí výběrové techniky, která se zaměřuje především na parametry *Priority* - priorita cílového hostitele a *Weight* - relativní váha pro záznamy se stejnou prioritou, se vybere server I-CSCF. Od chvíle, kdy pole doplňkových dat v odpovědi na výzvu obsahuje IP adresu vybraného serveru I-CSCF, tak již další DNS dotazy nejsou potřeba. Server P-CSCF přepošle žádost REGISTER na tuto IP adresu pomocí UDP protokolu a portu 5060.

Komunikace probíhá mezi porty 32770 a 53 na kterém je DNS server, viz. výpis níže.

```
|Time     | 127.0.0.1
|3.235    |       Standard query SRV DNS: Standard query SRV
                  _sip._udp.open-ims.test
|         |(32770)---------------------------------------> (53)
```

```
|3.235    |       Standard query respDNS: Standard query
                  response SRV 0 0 5060 icscf.open-ims.test
|         |(53)   ---------------------------------------> (32770)
```

c) zpráva REGISTER z P-CSCF do I-CSCF

Tato komunikace probíhá mezi portem 4060, jenž patří serveru P-CSCF a portem 5060 patřící serveru I-CSCF.

```
|Time     | 127.0.0.1
|3.236    |       Request: REGISTER sSIP: Request: REGISTER
                  sip:open-ims.test
|         |(4060) ---------------------------------------> (5060)
```

Změny oproti předchozí hlavičce:

```
REGISTER sip:open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK9f49.c3d86441.0
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bK2338ad9d23dfed589559019642261130
Max-Forwards: 16
Path: <sip:term@pcscf.open-ims.test:4060;lr>
Require: path
P-Visited-Network-ID: open-ims.test
P-Charging-Vector: icid-value="P-CSCFabcd4a0595ff00000000"; icid-
generated-at=127.0.0.1;orig-ioi="open-ims.test"
Authorization: Digest username="tom@open-ims.test", realm="open-
ims.test", nonce="", response="", uri="sip:open-ims.test",
integrity-protected="no"
```

Server P-CSCF musí být v poli *Path* pro všechny žádosti, které jsou směrovány k terminálu tohoto uživatele. P-CSCF tedy sám sebe přidá v hlavičce do pole *path* pro pozdější žádosti. Přidá také pole *P-Visited-Network-ID*, které obsahuje identifikátor P-CSCF sítě. To může být buď doménové jméno návštěvnické sítě nebo jakýkoliv jiný identifikátor, který ji identifikuje v domovské síti. P-CSCF také odstraní hlavičku *Security-Client* a sní spojený tag "*sec-agree*" před vlastním postoupením této žádosti. Jelikož hlavička *Proxy-Require* je prázdná, tak ji kompletně smaže.

Path - adresa P-CSCF serveru, slouží pro informování S-CSCF kam má směrovat žádosti.

Require - zajistí, aby příjemce správně zpracoval informace z hlavičky *Path*. Pokud příjemce nepodporuje hlavičku *Path*, tak budou odpovědi přijaty s kódem *420 Unsuported header*.

P-Visited-Network-ID - obsahuje identifikátor P-CSCF sítě v domácí síti.

P-Charging-Vector - P-CSCF vloží tohle záhlaví a naplní *icid* parametry globálně unikátníma hodnotama.

d) dotazy mezi I-CSCF a HSS: "Cx: User registration status query"

Cx-Query nebo ***Diameter UAR: User-Authorization-Request*** - server I-CSCF zažádá o informace týkající se stavu registrace účastníka zasláním soukromé identity uživatele, veřejné identity uživatele a identifikátoru návštěvnické sítě do HSS.

Cx-Query response nebo ***Diameter UAA: User-Authorization-Answer*** - HSS vrátí serveru S-CSCF požadované schopnosti a I-CSCF si pak podle těchto informací vybere vhodný S-CSCF server.

e) zpráva REGISTER z I-CSCF do S-CSCF

Komunikace probíhá mezi portem 5060, tedy serverem I-CSCF a portem 6060, na kterém máme server S-CSCF.

```
|Time     | 127.0.0.1
|3.444    |             Request: REGISTER sSIP: Request: REGISTER
                        sip:scscf.open-ims.test:6060
|         |(5060) ------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
REGISTER sip:scscf.open-ims.test:6060 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK9f49.e99e8407.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK9f49.c3d86441.0
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bK2338ad9d23dfed589559019642261130
Max-Forwards: 15
```

- I-CSCF nemění hodnotu pole *Path.*

P-Access-Network-Info - obsahuje informace od UE.
Path - S-CSCF si ukládá obsah pole *Path* a používá URI pro žádosti směrované do terminálu uživatele.

f) dotazy mezi S-CSCF a HSS: "Cx: Authentication"

Pokud přijde žádost REGISTER do P-CSCF bez kontroly integrity, tak by jí měl server S-CSCF odmítnout. Kvůli tomu S-CSCF požaduje použití alespoň jednoho autentizačního vektoru (AV) pro výzvu uživatele. Pokud není dostupný AV platný, pak server S-CSCF požaduje alespoň jeden AV z databázového serveru HSS. S-CSCF dále signalizuje HSS, že právě on byl vybrán k obsluze tohoto uživatele.

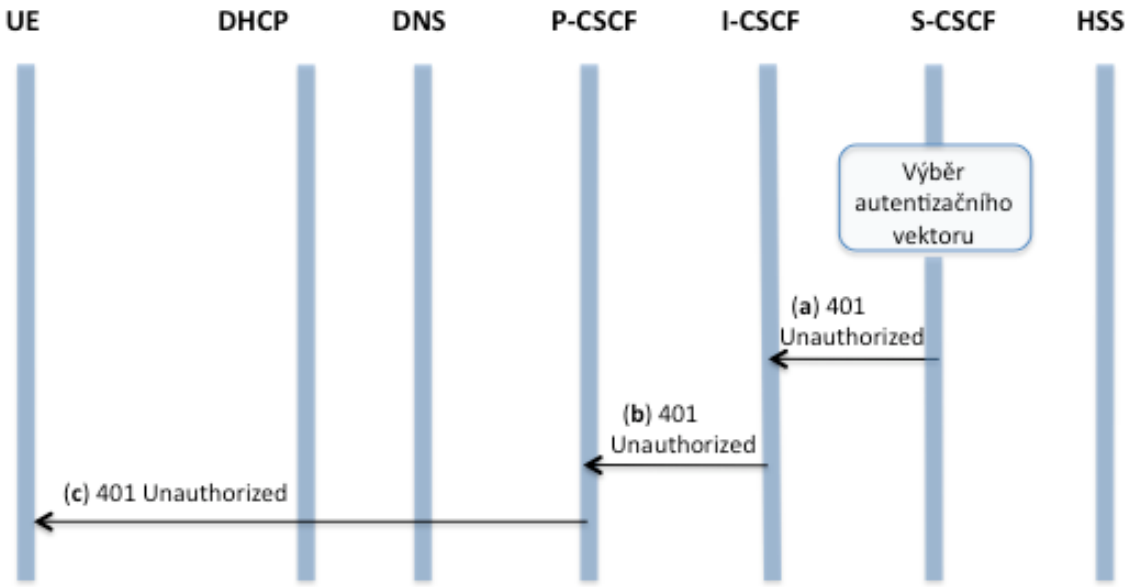
II. Zpráva 401 Unauthorized

S-CSCF vybere autentizační vektor pro výzvu k autentizaci. Tento AV by měl mít následující formu:

AV = RANDn | AUTNn | XRESn | CKn | IKn

kde

- RAND je náhodné číslo sloužící pro generování XRES, CK, IK a části AUTN. Používá se také ke generování RES v UE.
- AUTN je autentizační token
- XRES je očekávaná odpověď od UE
- CK je šifrovací klíč (volitelně)
- IK je klíč integrity



Obr. 6-5 Zaslání zprávy Unauthorized

a) zpráva 401 Unauthorized z S-CSCF do I-CSCF

Jak vidíme, tak komunikace probíhá mezi portem 6060, tedy serverem S-CSCF a portem 5060, na kterém máme server I-CSCF.

```
|Time     | 127.0.0.1
|3.578    |       Status: 401 UnauthoSIP: Status: 401
                  Unauthorized - Challenging the UE  (0 bindings)
|         |(6060) ------------------------------------> (5060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
SIP/2.0 401 Unauthorized - Challenging the UE
Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK9f49.e99e8407.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK9f49.c3d86441.0
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bK2338ad9d23dfed589559019642261130
From: "Tom" <sip:tom@open-ims.test>;tag=1000
To: "Tom" <sip:tom@open-ims.test>;
tag=d7837ce6bbd631122d10546eb75bb4cf-2deb
Call-ID: 5d6307d4f262aa6b00fb9321e0da2f68@127.0.0.1
CSeq: 1 REGISTER
WWW-Authenticate: Digest realm="open-ims.test",
nonce="4ac0f45bcca6a27887ec2121eaf6cdd1", algorithm=MD5,
qop="auth,auth-int"
Content-Length: 0
```

WWW-Authenticate - S-CSCF vyzývá uživatele. *Nonce* obsahuje řetězec, jenž představuje hodnotu složenou z AKA RAND, AKA AUTN a specifických dat serveru, a to vše zakódované pomocí base 64. V případě použití autentizace AKAv1-MD5 se zde přenáší i hodnota CK a IK.

b) zpráva 401 Unauthorized z I-CSCF do P-CSCF

Jak vidíme, tak komunikace probíhá mezi portem 5060, tedy serverem I-CSCF a portem 4060, na kterém máme server P-CSCF.

```
|Time     | 127.0.0.1
|3.580    |       Status: 401 UnauthoSIP: Status: 401
                  Unauthorized - Challenging the UE (0 bindings)
|         |(5060) ---------------------------------------> (4060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 401 Unauthorized - Challenging the UE
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK9f49.c3d86441.0
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bK2338ad9d23dfed589559019642261130
```

c) zpráva 401 Unauthorized z P-CSCF do UE

Komunikace zde probíhá mezi portem 4060, tedy serverem I-CSCF a portem 5061, na kterém máme našeho klienta.

```
|Time     | 127.0.0.1
|3.582    |       Status: 401 UnauthoSIP: Status: 401
                  Unauthorized - Challenging the UE (0 bindings)
|         |(4060) ---------------------------------------> (5061)
```

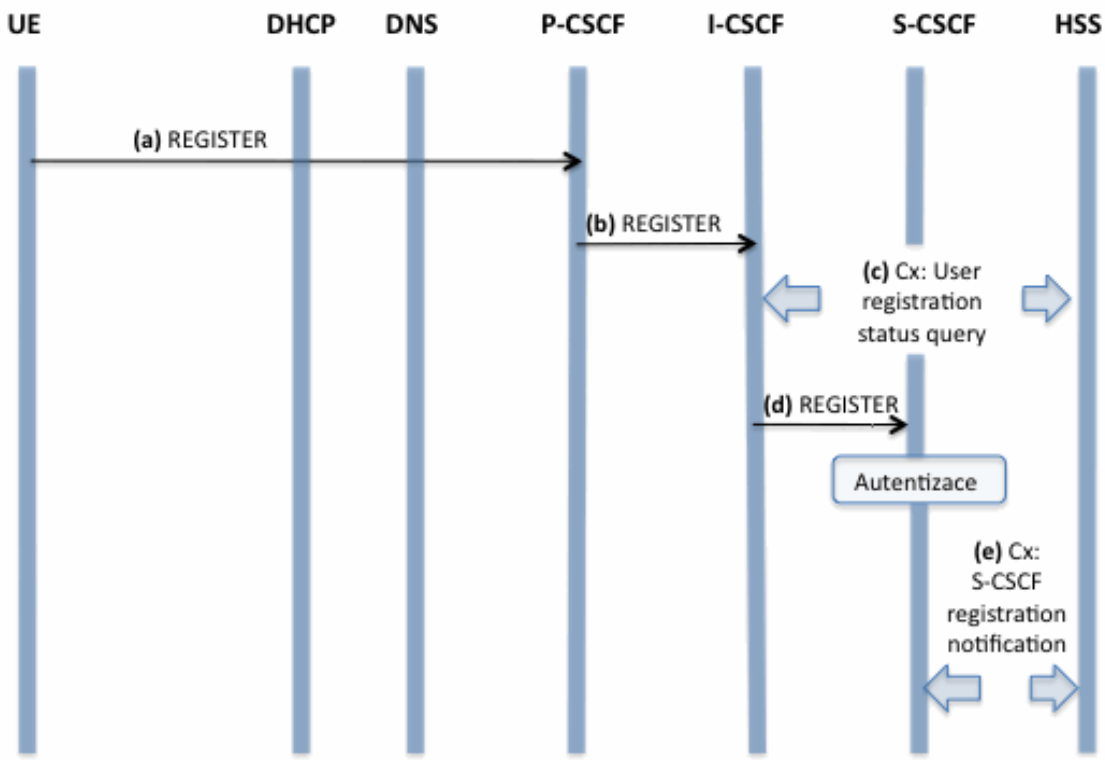
Změny oproti předchozí hlavičce:

```
SIP/2.0 401 Unauthorized - Challenging the UE
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bK2338ad9d23dfed589559019642261130
WWW-Authenticate: Digest realm="open-ims.test",
nonce="4ac0f45bcca6a27887ec2121eaf6cdd1", algorithm=MD5,
qop="auth,auth-int"
```

P-CSCF odstraní veškeré klíče, které obdržel v odpovědi *401 Unauthorized* a přepošle pak tuto odpověď do UE.

WWW-Authenticate - server P-CSCF v případě využití MKAv1-MD5 odstraní z hlavičky parametry CK a IK.

III. Zpráva REGISTER



Obr. 6-6 Zaslání zprávy REGISTER - druhá fáze

a) zpráva REGISTER z UE do P-CSCF

Komunikace mezi portem 5061, tedy naším klientem a portem 4060, serverem P-CSCF.

```
|Time      | 127.0.0.1
|3.602     |       Request: REGISTER sSIP: Request: REGISTER
                   sip:open-ims.test
|          |(5061) ------------------------------------> (4060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
REGISTER sip:open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5061;
branch=z9hG4bKf1c40a03f3dbac105497dbc0f6f99753
Max-Forwards: 20
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=00000000
From: "Tom" <sip:tom@open-ims.test>;tag=1001
To: "Tom" <sip:tom@open-ims.test>
Contact: <sip:127.0.0.1:5061>
Call-ID: 5d6307d4f262aa6b00fb9321e0da2f68@127.0.0.1
Authorization: Digest username="tom@open-ims.test", realm="open-
ims.test", nonce="4ac0f45bcca6a27887ec2121eaf6cdd1"
uri="sip:open-    ims.test", algorithm=MD5,
response="8aacf911bef4a0be68815303980d6f0a", qop=auth-int,
nc=00000001, cnonce="49529748544948102"
CSeq: 2 REGISTER
Supported: path
Content-Length: 0
```

Authorization - nese odpověď na dříve obdrženou výzvu k autentizaci spolu se soukromou identitou a s parametry *realm*, *nonce*, *URI* a *algorithm*.

b) zpráva REGISTER z P-CSCF do I-CSCF

Komunikace mezi portem 4060, tedy serverem P-CSCF a portem 5060, serverem I-CSCF.

```
|Time     | 127.0.0.1
|3.603    |       Request: REGISTER sSIP: Request: REGISTER
                  sip:open-ims.test
|         |(4060) ---------------------------------------> (5060)
```

Změny oproti předchozí hlavičce:

```
REGISTER sip:open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK9f49.c3d86441.0
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bKf1c40a03f3dbac105497dbc0f6f99753
Max-Forwards: 16
Path: <sip:term@pcscf.open-ims.test:4060;lr>
Require: path
P-Visited-Network-ID: open-ims.test
P-Charging-Vector: icid-value="P-CSCFabcd4a0595ff00000000"; icid-
generated-at=127.0.0.1;orig-ioi="open-ims.test"
Authorization: Digest username="tom@open-ims.test", realm="open-
ims.test", nonce="4ac0f45bcca6a27887ec2121eaf6cdd1" uri="sip:open-
ims.test", algorithm=MD5,
response="8aacf911bef4a0be68815303980d6f0a", qop=auth-int,
nc=00000001, cnonce="49529748544948102",integrity-protected="no"
```

c) dotazy mezi I-CSCF a HSS: "Cx: User registration status query"

Server I-CSCF žádá informace spojené se statusem registrace účastníka zasláním soukromé identity uživatele, veřejné identity a doménového jména návštěvnické sítě s HSS. Databázový server HSS navrací jméno serveru S-CSCF, který byl zvolen v kroku I. d).

d) zpráva REGISTER z I-CSCF do S-CSCF

Komunikace mezi portem 5060, tedy serverem I-CSCF a portem 6060, serverem S-CSCF.

```
|Time     | 127.0.0.1
|3.630    |       Request: REGISTER sSIP: Request: REGISTER
                  sip:scscf.open-ims.test:6060
|         |(5060) ---------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
REGISTER sip:open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK6f49.b04e47f4.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK9f49.c3d86441.0
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bKf1c40a03f3dbac105497dbc0f6f99753
Max-Forwards: 15
```

e) dotazy mezi S-CSCF a HSS: "Cx: S-CSCF registration notification"

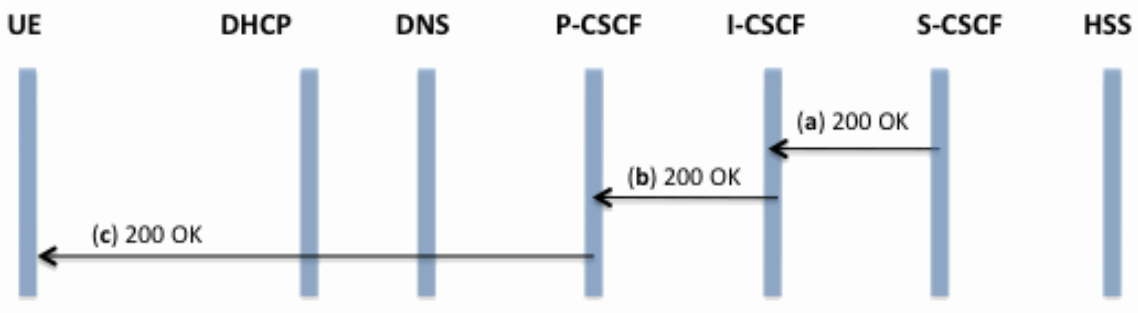
Autentizace - S-CSCF po obdržení žádosti REGISTER, která obsahuje odpověď na výzvu k autentizaci, zkontroluje, zda se očekávaná odpověď (vypočtená v S-CSCF pomocí XRES a dalších potřebných parametrů) shoduje s obdrženou odpovědí na výzvu k autentizaci. V případě shody je uživatel autentizován a jeho veřejná identita je registrována v S-CSCF.

Cx put nebo ***Diameter SAR: Server-Assignment-Request*** - při registraci uživatele server S-CSCF informuje HSS, že je uživatel od tohoto okamžiku registrován.

Cx put response nebo ***Diameter SAA: Server-Assignment-Answer*** - HSS po žádosti od S-CSCF zahrne profil uživatele do odpovědi zaslané do S-CSCF.

IV. Zpráva 200 OK

Tato signalizace značí, že registrace uživatele proběhla v pořádku.



Obr. 6-7 Zaslání zprávy 200 OK

a) zpráva 200 OK z S-CSCF do I-CSCF

Komunikace mezi portem 6060, tedy serverem S-CSCF a portem 5060, serverem I-CSCF.

```
|Time     | 127.0.0.1
|3.768    |             Status: 200 OK - SASIP: Status: 200 OK -
                 SAR succesful and registrar saved (1 bindings)
|         |(6060) ----------------------------------------> (5060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
SIP/2.0 200 OK - SAR succesful and registrar saved
Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK6f49.b04e47f4.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK6f49.585f9d61.0
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bKf1c40a03f3dbac105497dbc0f6f99753
Path: <sip:term@pcscf.open-ims.test:4060;lr>
Service-Route: <sip:orig@scscf.open-ims.test:6060;lr>
From: "Tom" <sip:tom@open-ims.test>;tag=1001
To: "Tom" <sip:tom@open-ims.test>;
tag=d7837ce6bbd631122d10546eb75bb4cf-8ae0
Call-ID: 5d6307d4f262aa6b00fb9321e0da2f68@127.0.0.1
Contact: <sip:127.0.0.1:5061>;expires=600
CSeq: 2 REGISTER
```

```
P-Associated-URI: <sip:tom@open-ims.test>
Content-Length: 0
```

Server S-CSCF zašle do I-CSCF odpověď *200 OK*, značící, že registrace proběhla v pořádku.

Service-Route - S-CSCF vloží pole *Service-Header*, včetně vlastního URI obsahujícího v uživatelské části řetězec znaků k rozlišení žádosti od UE a žádostí do UE.

b) zpráva 200 OK z I-CSCF do P-CSCF

Komunikace mezi portem 5060, tedy serverem I-CSCF a portem 4060, serverem P-CSCF.

```
|Time     | 127.0.0.1
|3.769    |             Status: 200 OK - SASIP: Status: 200 OK -
                    SAR succesful and registrar saved (1 bindings)
|         |(5060) ---------------------------------------> (4060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK - SAR succesful and registrar saved
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK6f49.585f9d61.0
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
```

I-CSCF přeposílá z S-CSCF do P-CSCF odpověď *200 OK*, značící, že registrace proběhla v pořádku. P-CSCF si uloží hodnotu pole *Service-Route* a spojí si ho s UE.

c) zpráva 200 OK z P-CSCF do UE

Komunikace mezi portem 4060, tedy serverem P-CSCF a portem 5061, patřící klientovi.

```
|Time     | 127.0.0.1
|3.770    |             Status: 200 OK - SASIP: Status: 200 OK -
SAR                     succesful and registrar saved   (1
bindings)
|         |(4060) ---------------------------------------> (5061)
```

Změny oproti předchozí hlavičce:

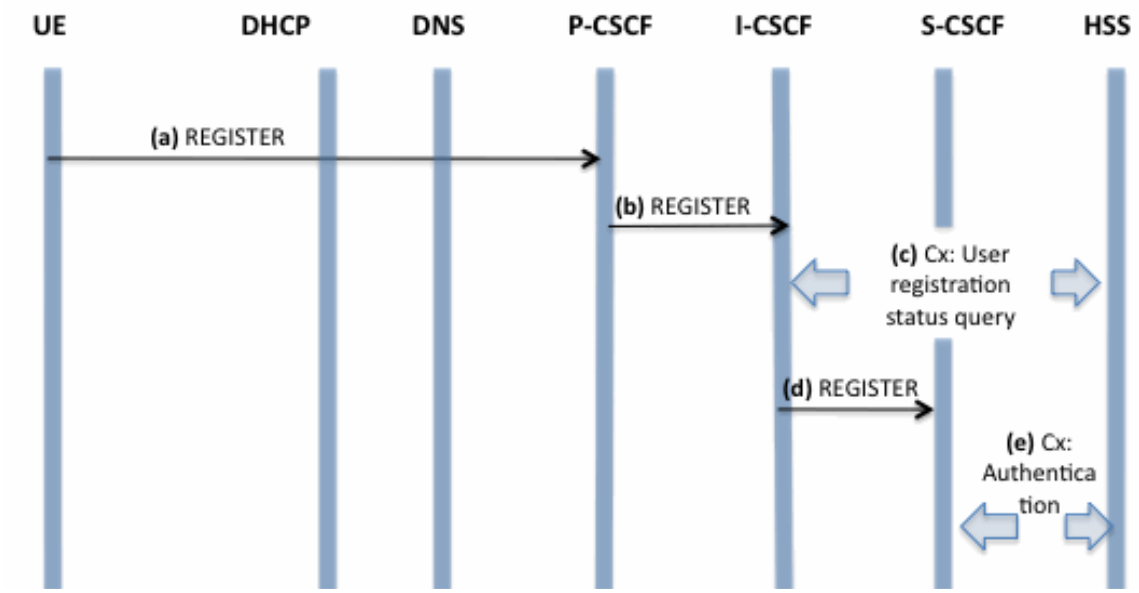
```
SIP/2.0 200 OK - SAR succesful and registrar saved
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
```

Odregistrování uživatele "Tom":

I. Zpráva REGISTER

Tato přenášená signalizace předpokládá stejný PDP kontext jako při počáteční registraci. Nyní se už nevyžaduje DHCP procedura pro nalezení P-CSCF a také už zde není potřeba výběr serveru S-CSCF pomocí I-CSCF.

UE se hodlá odregistrovat. To provede pomocí nové žádosti REGISTER, nyní však s polem v hlavičce Expires nastaveném na 0. Tato žádost je zaslána stejnému serveru P-CSCF, který dříve uživatele registroval.



Obr. 6-8 Zaslání zprávy REGISTER při odregistrování

a) zpráva REGISTER z UE do P-CSCF

Komunikace probíhá na portu 5061, jenž představuje port našeho klienta a na portu 4060, na kterém najdeme server P-CSCF.

```
|Time     | 127.0.0.1
|15.859   |             Request: REGISTER sSIP: Request: REGISTER
                        sip:open-ims.test
|         |(5061) ---------------------------------------> (4060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
REGISTER sip:open-ims.test SIP/2.0
Via: SIP/2.0/UDP
127.0.0.1:5061;branch=z9hG4bK8f5652907185ffd662141d88a7d30a9f
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=00000000
Max-Forwards: 20
From: "Tom" <sip:tom@open-ims.test>;tag=1003
To: "Tom" <sip:tom@open-ims.test>
Contact: <sip:127.0.0.1:5061>
Call-ID: 5d6307d4f262aa6b00fb9321e0da2f68@127.0.0.1
Authorization: Digest username="tom@open-ims.test",realm="open-
ims.test", nonce="b551c235849d4a41739bddc2c9aa72a9",uri="sip:open-
```

```
ims.test", algorithm=MD5,
response="203e4a97b19c122d79f260885ff92c96" qop=auth-int,
nc=00000001, cnonce="485110110048555652"
CSeq: 4 REGISTER
Supported: path
Content-Length: 0
```

Request-URI - následuje za názvem metody ("REGISTER") v prvním řádku, značí cílovou doménu této žádosti REGISTER. Pravidla pro směrování SIP žádostí popisují, jak využít DNS k rozpoznání adresy nebo vstupního bodu domovské sítě operátora (I-CSCF) doménového jména. Tato informace je uložena v USIM.

Via - adresa UE přidělená během procesu aktivace PDP kontextu.

P-Access-Network - UE poskytuje informace o přístupové metodě vztažené k obsluhující síti.

From - značí veřejnou identitu uživatele, který vyslal žádost REGISTER. Veřejná identita může být získána z USIM.

To - značí veřejnou identitu uživatele, který se registruje. Je to identita, podle které ostatní strany poznají tohoto účastníka.

Contact - označuje účastníka, IP adresu UE. Jedná se o dočasné označení pro účastníka, který se odregistrovává. Nulová hodnota pole *expires* značí, že se registrace uživatele ruší.

Authorization - nese autentizační informace. Soukromá identita uživatele (tom@open-ims.test) je uložena v Digest AKA protokolu v poli *username*. Proces odregistrování obsahuje také uložené informace jako - *realm, nonce, algorithm, uri* a *response*.

Supported - tohle záhlaví je zahrnuto v upozornění příjemce, že UE podporuje hlavičku Path.

b) zpráva REGISTER z P-CSCF do I-CSCF

Komunikace probíhá na portu 4060, jenž patří serveru P-CSCF a na portu 5060, na kterém najdeme server I-CSCF.

```
|Time     | 127.0.0.1
|15.860   |             Request: REGISTER sSIP: Request: REGISTER
                        sip:open-ims.test
|         |(4060) ---------------------------------------> (5060)
```

Změny oproti předchozí hlavičce:

```
REGISTER sip:open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK4f49.cee9308.0
Via: SIP/2.0/UDP 127.0.0.1:5061;
branch=z9hG4bK8f5652907185ffd662141d88a7d30a9f
Max-Forwards: 16
Path: <sip:term@pcscf.open-ims.test:4060;lr>
Require: path
P-Visited-Network-ID: open-ims.test
Authorization: Digest username="tom@open-ims.test",realm="open-
ims.test",nonce="b551c235849d4a41739bddc2c9aa72a9",uri="sip:open-
ims.test",algorithm=MD5,response="203e4a97b19c122d79f260885ff92c96
",qop=auth-int,nc=00000001,cnonce="485110110048555652", integrity-
protected="no"
```

P-CSCF odstraní hlavičku *Security-Client* a sní spojený tag "*sec-agree*" před vlastním postoupením této žádosti. Jelikož hlavička *Proxy-Require* je prázdná, tak ji kompletně smaže.

Path - adresa P-CSCF serveru, slouží pro informování S-CSCF kam má směrovat žádosti.

Require - zajistí, aby příjemce správně zpracoval informace z hlavičky *Path*. Pokud příjemce nepodporuje hlavičku Path, tak budou odpovědi přijaty s kódem *420 Unsuported header*.

P-Visited-Network-ID - obsahuje identifikátor P-CSCF sítě v domácí síti.

c) dotazy mezi I-CSCF a HSS: "Cx: User registration status query"

Cx-Query nebo ***Diameter UAR: User-Authorization-Request*** - I-CSCF zažádá o informace týkající se stavu registrace účastníka zasláním soukromé identity uživatele, veřejné identity uživatele a identifikátoru návštěvnické sítě do HSS.

Cx-Query response nebo ***Diameter UAA: User-Authorization-Answer*** - HSS vrátí serveru S-CSCF požadované schopnosti a I-CSCF si pak podle těchto informací vybere vhodný S-CSCF server.

d) zpráva REGISTER z I-CSCF do S-CSCF

Komunikace probíhá mezi porty 5060, tedy serverem I-CSCF a 6060, na kterém máme server S-CSCF.

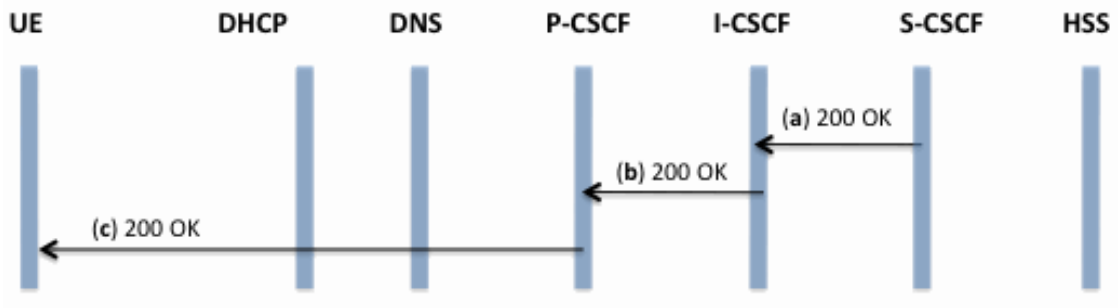
```
|Time     | 127.0.0.1
|15.891   |             Request: REGISTER sSIP: Request: REGISTER
                      sip:scscf.open-ims.test:6060
|         |(5060) ---------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
REGISTER sip:open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK4f49.058c0ba6.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK4f49.cee9308.0
Via: SIP/2.0/UDP 127.0.0.1:5061;
branch=z9hG4bK8f5652907185ffd662141d88a7d30a9f
Max-Forwards: 15
Path: <sip:term@pcscf.open-ims.test:4060;lr>
```

II. Zpráva 200 OK

Tato signalizace značí, že odregistrování uživatele proběhlo v pořádku.



Obr. 6-9 Zaslání zprávy 200 OK při odregistrování

a) zpráva 200 OK z S-CSCF do I-CSCF

Komunikace probíhá mezi porty 6060, tedy serverem S-CSCF a 5060, na kterém máme server I-CSCF.

```
|Time     | 127.0.0.1
|15.916   |            Status: 200 OK - SASIP: Status: 200 OK -
                    SAR succesful and registrar saved (1 bindings)
|         |(6060) ---------------------------------------> (5060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
SIP/2.0 200 OK - SAR succesful and registrar saved
Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK4f49.058c0ba6.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK4f49.cee9308.0
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bK8f5652907185ffd662141d88a7d30a9f
Path: <sip:term@pcscf.open-ims.test:4060;lr>
Service-Route: <sip:orig@scscf.open-ims.test:6060;lr>
From: "Tom" <sip:tom@open-ims.test>;tag=1003
To: "Tom" <sip:tom@open-
ims.test>;tag=d7837ce6bbd631122d10546eb75bb4cf-f25f
Call-ID: 5d6307d4f262aa6b00fb9321e0da2f68@127.0.0.1
Contact: <sip:127.0.0.1:5061>;expires=0
CSeq: 4 REGISTER
P-Associated-URI: <sip:tom@open-ims.test>
Content-Length: 0
```

b) zpráva 200 OK z I-CSCF do P-CSCF

Komunikace probíhá mezi porty 5060, tedy serverem I-CSCF a 4060, na kterém máme server P-CSCF.

```
|Time     | 127.0.0.1
|15.917   |            Status: 200 OK - SASIP: Status: 200 OK -
                    SAR succesful and registrar saved (1 bindings)
|         |(5060) ---------------------------------------> (4060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK - SAR succesful and registrar saved
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK4f49.cee9308.0
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bK8f5652907185ffd662141d88a7d30a9f
Service-Route: <sip:orig@scscf.open-ims.test:6060;lr>
```

c) zpráva 200 OK z P-CSCF do UE

Komunikace probíhá mezi porty 4060, tedy serverem P-CSCF a portem 5061, na kterém máme našeho klienta.

```
|Time     | 127.0.0.1
|15.919   |             Status: 200 OK - SASIP: Status: 200 OK -
                  SAR succesful and registrar saved (1 bindings)
|         |(4060)   ------------------------------------->(5061)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK - SAR succesful and registrar saved
Via: SIP/2.0/UDP 127.0.0.1:5061; rport=5061;
branch=z9hG4bK8f5652907185ffd662141d88a7d30a9f
```

6.2 Sestavení, průběh a ukončení spojení

Nyní si ukážeme jak vypadá průběh ustálení spojení při volání mezi Bobem (UE1) na portu 5061 a Alicí (UE2) na portu 5062, jenž se oba nachází v domovské síti. Je vytvořeno RTP (Real-time Transport Protocol) spojení na UDP portech 8001 pro Alici a 8000 pro Boba. Následující průběh odpovídá situaci, kdy Bob volá Alici (zpráva *INVITE*), ta hovor přijme (zpráva *180 Ringing*) a dochází k přenosu hovorových dat. Poté Alice hovor ukončí (zpráva *Bye*).

UE1 se tedy nachází v domovské síti a k rozpoznání serveru P-CSCF použije procedury k nalezení CSCF. Server P-CSCF už zná jméno (adresu) serveru S-CSCF díky předchozí registraci UE1.

[12, 14]

I. Zpráva INVITE z UE1 (Bob) do UE2 (Alice)

Žádost o navázání spojení INVITE v sobě nese i zapouzdřenou zpráva protokolu SDP (Session Description Protocol), jenž nám specifikuje použité kódování pro multimediální data, jejich parametry a čísla portů pro spojení.

a) zpráva INVITE z UE1 (Bob) do serveru P-CSCF

Komunikace probíhá mezi portem 5061, který obsadil UE1 (Bob) a portem 4060, tedy serverem P-CSCF. V opačném směru se zasílá odpověď *100 trying*, značící zpracování přijaté žádosti.

```
|Time     | 127.0.0.1
|18.674   |         Request: INVITE sipSIP/SDP: Request: INVITE
              sip:alice@open-ims.test, with session description
|         |(5061) ---------------------------------------> (4060)

|18.675   |         Status: 100 trying SIP: Status: 100 trying --
                        your call is important to us
|         |(4060) ---------------------------------------> (5061)
```

Hlavička protokolu SIP zde vypadá následovně:

```
INVITE sip:alice@open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5061;
branch=z9hG4bK2d9bb268e22f1715bf9008ac93c346ba
Max-Forwards: 20
Route: <sip:orig@scscf.open-ims.test:6060;lr>
P-Preferred-Identity: "Bob" <sip:bob@open-ims.test>
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=00000000
Privacy: none
From: "Bob" <sip:bob@open-ims.test>;tag=1002
To: <sip:alice@open-ims.test>
Call-ID: 9e84d1d574e8938ce330b792879cb6be@127.0.0.1
CSeq: 3 INVITE
Contact: <sip:127.0.0.1:5061>
Allow: INVITE,ACK,CANCEL,BYE,MESSAGE,NOTIFY
Content-Type: application/sdp
Content-Length: 113
```

```
SDP
v=0 - SDP version
o=user 0 0 IN IP4 127.0.0.1 - (vlastník: User, ID relace: 0, verze
relace : 0, typ sítě: IN, typ adresy: IP4, adresa: 127.0.0.1)
s=The funky IMS stream - (jméno relace)
c=IN IP4 127.0.0.1 - (informace o spojení - typ sítě: IN, typ
adresy:     IP4, adresa: 127.0.0.1)
t=0 0 -  ( čas začátku: 0, čas konce: 0)
m=audio 8000 RTP/AVP 0 3 8 - (typ dat: audio, port: 8000,
protokol:   RTP/AVP, formát dat: 0=ITU-T G.711 PCMU, 3=GSM 06.10,
8=ITU-T G.711     PCMA)
```

UE1 vytvoří SDP nabídku, která obsahuje požadavky na šířku pásma a vlastosti všech kodeků, které je možné pro tuto relaci použít. Každému mediálnímu toku přiřadí zdejší čísla portů. UE1 je ochoten vytvořit multimediální relaci zahrnující audio i video přenos. Pro video jsou k dispozici dva kodeky, buď H.263 nebo MPEG-4. Pro audio zde máme kodek AMR.

Request-URI - obsahuje číslo uživatele v mezinárodním tvaru podle doporučení ITU E.164.

Via - obsahuje buď IP adresu nebo jméno domény UE, ze kterého tato zpráva pochází.

Route - obsahuje adresu serveru S-CSCF s číslem portu a části hlavičky *Service-Route* z registrace.

P-Preferred-Identity - uživatel naznačuje, která identita se má použít pro tuto relaci.

P-Access-Network-Info - uživatel poskytuje informace o přístupové síti a typ přístupové metody.

Privacy - uživatel nevyžaduje utajení dat, proto má tato hlavička hodnotu *none*.

b) zpráva INVITE z P-CSCF do S-CSCF

Komunikace probíhá mezi portem 5061, který obsadil UE1 (Bob) a portem 4060, tedy serverem P-CSCF. V opačném směru se zasílá odpověď *100 trying*, značící zpracování přijaté žádosti.

```
|Time     | 127.0.0.1
|18.675   |         Request: INVITE sipSIP/SDP: Request: INVITE
              sip:alice@open-ims.test, with session description
|         |(4060) ---------------------------------------> (6060)

|18.677   |         Status: 100 trying SIP: Status: 100 trying --
                      your call is important to us
|         |(6060) ---------------------------------------> (4060)
```

Změny oproti předchozí hlavičce:

```
INVITE sip:alice@open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK69bf.02a1cf94.0
Max-Forwards: 16
Route: <sip:orig@scscf.open-ims.test:6060;lr>
Record-Route: <sip:mo@pcscf.open-ims.test:4060;lr>
P-Asserted-Identity: "Bob" <sip:bob@open-ims.test>
P-Charging-Vector: icid-value="P-CSCFabcd4a11571c00000015";icid-
generated-at=127.0.0.1;orig-ioi="open-ims.test"
```

Server P-CSCF sáme sebe přidá do hlavičky *Record-Route* a *Via*.

P-Asserted-Identity - server P-CSCF vloží do této hlavičky parametr URI a odstraní hlavičku *P-Preferred-Identity.*

P-Charging-Vector - P-CSCF vloží tohle záhlaví a naplní *icid* parametry globálně unikátníma hodnotama.

Server S-CSCF ověří profil služeb tohoto účastníka a vyhodnocuje základní kritéria pro filtraci.

c) zpráva INVITE z S-CSCF do S-CSCF

Jelikož v naší síti oba uživatelé využívají služeb stejného serveru S-CSCF, tak neprobíhá komunikace mezi serverem I-CSCF pro případné zjištění adresy serveru S-CSCF. Zároveň pak odpadá komunikace mezi I-CSCF a HSS, kde by došlo k výměně zpráv protokolu Diameter LIR a LIA (Location-Info-Request/Answer). Zpráva INVITE se tak zašle přímo na S-CSCF server druhého uživatele. Komunikace tedy probíhá na portu 6060, kdy si server S-CSCF sám sobě přepošle zprávu *INVITE*. V opačném směru se zasílá odpověď *100 trying*, značící zpracování přijaté žádosti.

```
|Time     | 127.0.0.1
|18.677   |         Request: INVITE sipSIP/SDP: Request: INVITE
                sip:alice@open-ims.test, with session description
|         |(6060) ---------------------------------------> (6060)

|18.678   |         Status: 100 trying SIP: Status: 100 trying --
                      your call is important to us
|         |(6060) ---------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
INVITE sip:alice@open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:6060;branch=z9hG4bK69bf.c56f9902.0
Max-Forwards: 15
Record-Route: <sip:mo@scscf.open-ims.test:6060;lr>
```

Server P-CSCF sám sebe přidá do hlavičky *Record-Route* a *Via*.

d) zpráva INVITE z S-CSCF do P-CSCF

Komunikace probíhá mezi portem 6060, patřící serveru S-CSCF a portem 4060, tedy serverem P-CSCF. V opačném směru se zasílá odpověď *100 trying*, značící zpracování přijaté žádosti.

```
|Time     | 127.0.0.1
|18.680   |         Request: INVITE sipSIP/SDP: Request: INVITE
                  sip:127.0.0.1:5062, with session description
|         |(6060) ---------------------------------------> (4060)

|18.680   |         Status: 100 trying SIP: Status: 100 trying --
                      your call is important to us
|         |(4060) ---------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
INVITE sip:alice@open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:6060;branch=z9hG4bK69bf.d56f9902.0
Via: SIP/2.0/UDP 127.0.0.1:6060;branch=z9hG4bK69bf.c56f9902.0
Max-Forwards: 14
Record-Route: <sip:mt@scscf.open-ims.test:6060;lr>
Route: <sip:term@pcscf.open-ims.test:4060;lr>
P-Called-Party-ID: <sip:alice@open-ims.test>
```

Server S-CSCF si z registrace uživatele UE2 pamatuje jeho kontaktní adresu a jeho další CSCF server.

Route - je vytvořena z hlavičky *Path* z registrace.
Via/Record-Route - zde server S-CSCF přidá o sobě záznam.
P-Called-Party-ID - obsahuje vytáčené URL s jeho parametry.

e) zpráva INVITE z P-CSCF do UE2 (Alice)

Komunikace probíhá mezi portem 6060, patřící serveru S-CSCF a portem 4060, tedy serverem P-CSCF.

```
|Time     | 127.0.0.1
|18.680   |       Request: INVITE sipSIP/SDP: Request: INVITE
                  sip:127.0.0.1:5062, with session description
|         |(4060) ---------------------------------------> (5062)
```

Změny oproti předchozí hlavičce:

```
INVITE sip:alice@open-ims.test SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK69bf.12a1cf94.0
Max-Forwards: 13
Record-Route: <sip:mt@pcscf.open-ims.test:4060;lr>
```

II. Zpráva 180 Ringing z UE2 (Alice) do UE1 (Bob)

Tato přenášená zpráva je odpovědí z UE na zprávu INVITE a signalizuje vyzvánění volaného.

a) zpráva 180 Ringing z UE2 (Alice) do P-CSCF

Komunikace probíhá mezi portem 5062, na kterém máme klienta UE2 a portem 4060, tedy serverem P-CSCF.

```
|Time     | 127.0.0.1
|18.691   |         Status: 180 RingingSIP: Status: 180 Ringing
|         |(5062) ---------------------------------------> (4060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK69bf.12a1cf94.0,SIP/2.0/UDP
127.0.0.1:6060;rport=6060;branch=z9hG4bK69bf.d56f9902.0,SIP/2.0/
UDP
```

```
127.0.0.1:6060;branch=z9hG4bK69bf.c56f9902.0,SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK69bf.02a1cf94.0,SIP/2.0/UDP
127.0.0.1:5061;rport=5061;branch=z9hG4bK2d9bb268e22f1715bf9008ac93
c346ba
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=00000000
From: "Bob" <sip:bob@open-ims.test>;tag=1002
To: <sip:alice@open-ims.test>
Call-ID: 9e84d1d574e8938ce330b792879cb6be@127.0.0.1
CSeq: 3 INVITE
Contact: <sip:127.0.0.1:5062>
Content-Length: 0
```

Před ustálením spojení musí UE2 ještě vyčkat na úspěšnou rezervaci prostředků pro přenos. Poté může oznámit koncovému účastníkovi přicházející spojení. To provede zasláním do serveru P-CSCF směrem k volající straně zprávou *180 Ringing*.

b) zpráva 180 Ringing z P-CSCF do S-CSCF

Komunikace probíhá mezi portem 4060, patřící serveru P-CSCF a portem 6060, tedy serverem S-CSCF.

```
|Time     | 127.0.0.1
|18.692   |         Status: 180 RingingSIP: Status: 180 Ringing
|         |(4060) ------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP
127.0.0.1:6060;rport=6060;branch=z9hG4bK69bf.d56f9902.0,SIP/2.0/
UDP
127.0.0.1:6060;branch=z9hG4bK69bf.c56f9902.0,SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK69bf.02a1cf94.0,SIP/2.0/UDP
127.0.0.1:5061;rport=5061;branch=z9hG4bK2d9bb268e22f1715bf9008ac93
c346ba
```

c) zpráva 180 Ringing z S-CSCF do S-CSCF

Komunikace tedy probíhá na portu 6060, kdy si server S-CSCF sám sobě přepošle zprávu *180 Ringing*.

```
|Time     | 127.0.0.1
|18.693   |         Status: 180 RingingSIP: Status: 180 Ringing
|         |(6060) ------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP
127.0.0.1:6060;branch=z9hG4bK69bf.c56f9902.0,SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK69bf.02a1cf94.0,SIP/2.0/UDP
127.0.0.1:5061;rport=5061;branch=z9hG4bK2d9bb268e22f1715bf9008ac93
c346ba
```

d) zpráva 180 Ringing z S-CSCF do P-CSCF

Komunikace probíhá mezi portem 6060, patřící serveru S-CSCF a portem 4060, tedy serverem P-CSCF.

```
|Time     | 127.0.0.1
|18.693   |         Status: 180 RingingSIP: Status: 180 Ringing
|         |(6060) ---------------------------------------> (4060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK69bf.02a1cf94.0,SIP/2.0/UDP
127.0.0.1:5061;rport=5061;branch=z9hG4bK2d9bb268e22f1715bf9008ac93
c346ba
```

e) zpráva 180 Ringing z P-CSCF do UE1 (Bob)

Komunikace probíhá mezi portem 4060, patřící serveru P-CSCF a portem 5061, na kterém je uživatel UE1 (Bob).

```
|Time     | 127.0.0.1
|18.694   |         Status: 180 RingingSIP: Status: 180 Ringing
|         |(4060) ---------------------------------------> (5061)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP
127.0.0.1:5061;rport=5061;branch=z9hG4bK2d9bb268e22f1715bf9008ac93
c346ba
```

III. Zpráva 200 OK z UE2 (Alice) do UE1 (Bob)

Jakmile Alice přijme hovor od Boba, tak je směrem k Bobovi zaslána zpráva 200 OK, ve které je zapouzdřena zpráva protokolu SDP (Session Description Protocol), jenž nám specifikuje použité kódování pro multimediální data, jejich parametry a čísla portů.

a) zpráva 200 OK z UE2 (Alice) do P-CSCF

Komunikace probíhá mezi portem 5062, na kterém je uživatel UE2 (Alice) a portem 4060, patřící serveru P-CSCF.

```
|Time     | 127.0.0.1
|27.745   |         Status: 200 OK, witSIP/SDP: Status: 200 OK,
                        with session description
|         |(5062) ---------------------------------------> (4060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK69bf.12a1cf94.0,SIP/2.0/UDP
127.0.0.1:6060;rport=6060;branch=z9hG4bK69bf.d56f9902.0,SIP/2.0/UD
```

```
P 127.0.0.1:6060;branch=z9hG4bK69bf.c56f9902.0,SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK69bf.02a1cf94.0,SIP/2.0/UDP
127.0.0.1:5061;rport=5061;branch=z9hG4bK2d9bb268e22f1715bf9008ac93
c346ba
Record-Route: <sip:mt@pcscf.open-ims.test:4060;lr>,
<sip:mt@scscf.open-ims.test:6060;lr>, <sip:mo@scscf.open-
ims.test:6060;lr>, <sip:mo@pcscf.open-ims.test:4060;lr>
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=00000000
From: "Bob" <sip:bob@open-ims.test>;tag=1002
To: <sip:alice@open-ims.test>
Call-ID: 9e84d1d574e8938ce330b792879cb6be@127.0.0.1
CSeq: 3 INVITE
Contact: <sip:127.0.0.1:5062>
Content-Length: 113
```

Jakmile volaný účastník odpoví, tak UE2 pošle na žádost *INVITE* konečnou odpověď 200 OK do P-CSCF a začne tok dat pro tuto relaci. P-CSCF také schvaluje nasazení QoS.

b) zpráva 200 OK z P-CSCF do S-CSCF

Komunikace probíhá mezi portem 4060, tedy serverem P-CSCF a portem 6060, patřící serveru S-CSCF.

```
|Time     | 127.0.0.1
|27.747   |         Status: 200 OK, witSIP/SDP: Status: 200 OK,
                        with session description
|         |(4060) ---------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.0.1:6060;rport=6060;branch=z9hG4bK69bf.d56f9902.0,SIP/2.0/
UDP
127.0.0.1:6060;branch=z9hG4bK69bf.c56f9902.0,SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK69bf.02a1cf94.0,SIP/2.0/UDP
127.0.0.1:5061;rport=5061;branch=z9hG4bK2d9bb268e22f1715bf9008ac93
c346ba
```

c) zpráva 200 OK z S-CSCF do S-CSCF

Komunikace tedy probíhá na portu 6060, kdy si server S-CSCF sám sobě přepošle zprávu *180 Ringing.*

```
|Time     | 127.0.0.1
|27.748   |         Status: 200 OK, witSIP/SDP: Status: 200 OK,
                        with session description
|         |(6060) ---------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.0.1:6060;branch=z9hG4bK69bf.c56f9902.0,SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK69bf.02a1cf94.0,SIP/2.0/UDP
```

```
127.0.0.1:5061;rport=5061;branch=z9hG4bK2d9bb268e22f1715bf9008ac93
c346ba
```

d) zpráva 200 OK z S-CSCF do P-CSCF

Komunikace probíhá mezi portem 6060, tedy serverem S-CSCF a portem 4060, patřící serveru P-CSCF.

```
|Time     | 127.0.0.1
|27.748   |         Status: 200 OK, witSIP/SDP: Status: 200 OK,
                            with session description
|         |(6060) ---------------------------------------> (4060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK69bf.02a1cf94.0,SIP/2.0/UDP
127.0.0.1:5061;rport=5061;branch=z9hG4bK2d9bb268e22f1715bf9008ac93
c346ba
```

e) zpráva 200 OK z P-CSCF do UE1 (Bob)

Komunikace probíhá mezi portem 4060, tedy serverem P-CSCF a portem 5061, na kterém je uživatel UE1 (Bob).

```
|Time     | 127.0.0.1
|27.750   |         Status: 200 OK, witSIP/SDP: Status: 200 OK,
                        with session description
|         |(4060) ---------------------------------------> (5061)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.0.1:5061;rport=5061;branch=z9hG4bK2d9bb268e22f1715bf9008ac93
c346ba
```

IV. Zpráva ACK z UE1 (Bob) do UE2 (Alice)

Zpráva ACK představuje potvrzení o obdržení odpovědi na žádost *INVITE*.

a) zpráva ACK z UE1 (Bob) do P-CSCF

Komunikace probíhá mezi portem 5061, na kterém je uživatel UE1 (Bob) a portem 4060, tedy serverem P-CSCF.

```
|Time     | 127.0.0.1
|27.758   |         Request: ACK sip:12SIP: Request: ACK
                        sip:127.0.0.1:5062
|         |(5061) ---------------------------------------> (4060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
ACK sip:127.0.0.1:5062 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5061;
branch=z9hG4bKc79fab8cfb0dc3f4b3b05aeb7644362b
Max-Forwards: 70
Route: <sip:mo@pcscf.open-ims.test:4060;lr>, <sip:mo@scscf.open-
ims.test:6060;lr>, <sip:mt@scscf.open-ims.test:6060;lr>,
<sip:mt@pcscf.open-ims.test:4060;lr>
From: "Bob" <sip:bob@open-ims.test>;tag=1002
To: <sip:alice@open-ims.test>
Call-ID: 9e84d1d574e8938ce330b792879cb6be@127.0.0.1
CSeq: 3 ACK
Content-Length: 0
```

b) zpráva ACK z P-CSCF do S-CSCF

Komunikace probíhá mezi portem 4060, tedy serverem P-CSCF a portem 6060, tedy serverem S-CSCF.

```
|Time     | 127.0.0.1
|27.759   |         Request: ACK sip:12SIP: Request: ACK
                        sip:127.0.0.1:5062
|         |(4060) ---------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
ACK sip:127.0.0.1:5062 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=0
Max-Forwards: 16
Route: <sip:mo@scscf.open-ims.test:6060;lr>,<sip:mt@scscf.open-
ims.test:6060;lr>,<sip:mt@pcscf.open-ims.test:4060;lr>
```

c) zpráva ACK z S-CSCF do S-CSCF

Komunikace tedy probíhá na portu 6060, kdy si server S-CSCF sám sobě přepošle zprávu *ACK*.

```
|Time     | 127.0.0.1
|27.761   |         Request: ACK sip:12SIP: Request: ACK
                        sip:127.0.0.1:5062
|         |(6060) ---------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
ACK sip:127.0.0.1:5062 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:6060;branch=0
Max-Forwards: 15
Route: <sip:mt@scscf.open-ims.test:6060;lr>,<sip:mt@pcscf.open-
ims.test:4060;lr>
```

d) zpráva ACK z S-CSCF do P-CSCF

Komunikace probíhá mezi portem 6060, tedy serverem S-CSCF a portem 4060, tedy serverem P-CSCF.

```
|Time     | 127.0.0.1
|27.761   |         Request: ACK sip:12SIP: Request: ACK
                        sip:127.0.0.1:5062
|         |(6060) ---------------------------------------> (4060)
```

Změny oproti předchozí hlavičce:

```
ACK sip:127.0.0.1:5062 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:6060;branch=0
Max-Forwards: 14
Route: <sip:mt@pcscf.open-ims.test:4060;lr>
```

d) zpráva ACK z P-CSCF do UE2 (Alice)

Komunikace probíhá mezi portem 4060, tedy serverem P-CSCF a portem 5062, na kterém je uživatel UE2 (Alice).

```
|Time     | 127.0.0.1
|27.762   |         Request: ACK sip:12SIP: Request: ACK
                        sip:127.0.0.1:5062
|         |(4060) ---------------------------------------> (5062)
```

Změny oproti předchozí hlavičce:

```
ACK sip:127.0.0.1:5062 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=0
Max-Forwards: 13
```

V. Zpráva BYE z UE2 (Alice) do UE1 (Bob)

Zpráva BYE představuje žádost o ukončení spojení. Zde Tedy Alice žádá o ukončení spojení.

a) zpráva BYE z UE2 (Alice) do P-CSCF

Komunikace probíhá mezi portem 5062, na kterém je uživatel UE2 (Alice) a portem 4060, tedy serverem P-CSCF.

```
|Time     | 127.0.0.1
|39.541   |         Request: BYE sip:12SIP: Request: BYE
                        sip:127.0.0.1:5061
|         |(5062) ---------------------------------------> (4060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
BYE sip:127.0.0.1:5061 SIP/2.0
Via: SIP/2.0/UDP
127.0.0.1:5062;branch=z9hG4bK5862405872337de669668bc9f511d12b
Max-Forwards: 70
```

```
Route: <sip:mt@pcscf.open-ims.test:4060;lr>, <sip:mt@scscf.open-
ims.test:6060;lr>, <sip:mo@scscf.open-ims.test:6060;lr>,
<sip:mo@pcscf.open-ims.test:4060;lr>
From: <sip:alice@open-ims.test>;tag=1014
To: "Bob" <sip:bob@open-ims.test>;tag=1002
Call-ID: 9e84d1d574e8938ce330b792879cb6be@127.0.0.1
CSeq: 1 BYE
Content-Length: 0
```

b) zpráva BYE z P-CSCF do S-CSCF

Komunikace probíhá mezi portem 4060, tedy serverem P-CSCF a portem 6060, tedy serverem S-CSCF.

```
|Time     | 127.0.0.1
|39.543   |         Request: BYE sip:12SIP: Request: BYE
                        sip:127.0.0.1:5061
|         |(4060) ------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
BYE sip:127.0.0.1:5061 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK89bf.37458eb.0
Max-Forwards: 16
Route: <sip:mt@scscf.open-ims.test:6060;lr>, <sip:mo@scscf.open-
ims.test:6060;lr>, <sip:mo@pcscf.open-ims.test:4060;lr>
```

c) zpráva BYE z S-CSCF do S-CSCF

Komunikace tedy probíhá na portu 6060, kdy si server S-CSCF sám sobě přepošle zprávu *BYE*.

```
|Time     | 127.0.0.1
|39.543   |         Request: BYE sip:12SIP: Request: BYE
                        sip:127.0.0.1:5061
|         |(6060) ------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
BYE sip:127.0.0.1:5061 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:6060;branch=z9hG4bK89bf.a2b4ea84.0
Max-Forwards: 15
Route: <sip:mo@scscf.open-ims.test:6060;lr>, <sip:mo@pcscf.open-
ims.test:4060;lr>
```

d) zpráva BYE z S-CSCF do P-CSCF

Komunikace probíhá mezi portem 6060, tedy serverem S-CSCF a portem 4060, tedy serverem P-CSCF.

```
|Time     | 127.0.0.1
|39.543   |         Request: BYE sip:12SIP: Request: BYE
                        sip:127.0.0.1:5061
|         |(6060) ------------------------------------> (4060)
```

Změny oproti předchozí hlavičce:

```
BYE sip:127.0.0.1:5061 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:6060;branch=z9hG4bK89bf.b2b4ea84.0
Max-Forwards: 14
Route: <sip:mo@pcscf.open-ims.test:4060;lr>
```

e) zpráva BYE z P-CSCF do UE1 (Bob)

Komunikace probíhá mezi portem 4060, tedy serverem P-CSCF a portem 5061, na kterém je uživatel UE1 (Bob).

```
|Time     | 127.0.0.1
|39.544   |         Request: BYE sip:12SIP: Request: BYE
                        sip:127.0.0.1:5061
|         |(4060) ---------------------------------------> (5061)
```

Změny oproti předchozí hlavičce:

```
BYE sip:127.0.0.1:5061 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:4060;branch=z9hG4bK89bf.47458eb.0
Max-Forwards: 13
```

V. Zpráva 200 OK z UE1 (Bob) do UE2 (Alice)

Zpráva 200 OK značí úspěšné provedení předchozí žádosti, tedy ukončení spojení mezi Alicí a Bobem.

a) zpráva 200 OK z UE1 (Bob) do P-CSCF

Komunikace probíhá mezi portem 5061, na kterém je uživatel UE1 (Bob) a portem 4060, tedy serverem P-CSCF.

```
|Time     | 127.0.0.1
|39.559   |         Status: 200 OK SIP: Status: 200 OK
|         |(5061) ---------------------------------------> (4060)
```

Hlavička protokolu SIP zde vypadá následovně:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK89bf.47458eb.0,SIP/2.0/UDP
127.0.0.1:6060;rport=6060;branch=z9hG4bK89bf.b2b4ea84.0,SIP/2.0/
UDP
127.0.0.1:6060;branch=z9hG4bK89bf.a2b4ea84.0,SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK89bf.37458eb.0,SIP/2.0/UDP
127.0.0.1:5062;rport=5062;branch=z9hG4bK5862405872337de669668bc9f5
11d12b
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=00000000
From: <sip:alice@open-ims.test>;tag=1014
To: "Bob" <sip:bob@open-ims.test>;tag=1002
Call-ID: 9e84d1d574e8938ce330b792879cb6be@127.0.0.1
CSeq: 1 BYE
Content-Length: 0
```

b) zpráva 200 OK z P-CSCF do S-CSCF

Komunikace probíhá mezi portem 4060, tedy serverem P-CSCF a portem 6060, tedy serverem S-CSCF.

```
|Time     | 127.0.0.1
|39.560   |         Status: 200 OK SIP: Status: 200 OK
|         |(4060) ---------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.0.1:6060;rport=6060;branch=z9hG4bK89bf.b2b4ea84.0,SIP/2.0/
UDP
127.0.0.1:6060;branch=z9hG4bK89bf.a2b4ea84.0,SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK89bf.37458eb.0,SIP/2.0/UDP
127.0.0.1:5062;rport=5062;branch=z9hG4bK5862405872337de669668bc9f5
11d12b
```

c) zpráva 200 OK z S-CSCF do S-CSCF

Komunikace tedy probíhá na portu 6060, kdy si server S-CSCF sám sobě přepošle zprávu *200 OK*.

```
|Time     | 127.0.0.1
|39.560   |         Status: 200 OK SIP: Status: 200 OK
|         |(6060) ---------------------------------------> (6060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.0.1:6060;branch=z9hG4bK89bf.a2b4ea84.0,SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK89bf.37458eb.0,SIP/2.0/UDP
127.0.0.1:5062;rport=5062;branch=z9hG4bK5862405872337de669668bc9f5
11d12b
```

d) zpráva 200 OK z S-CSCF do P-CSCF

Komunikace probíhá mezi portem 6060, tedy serverem S-CSCF a portem 4060, tedy serverem P-CSCF.

```
|Time     | 127.0.0.1
|39.560   |         Status: 200 OK SIP: Status: 200 OK
|         |(6060) ---------------------------------------> (4060)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.0.1:4060;branch=z9hG4bK89bf.37458eb.0,SIP/2.0/UDP
127.0.0.1:5062;rport=5062;branch=z9hG4bK5862405872337de669668bc9f5
11d12b
```

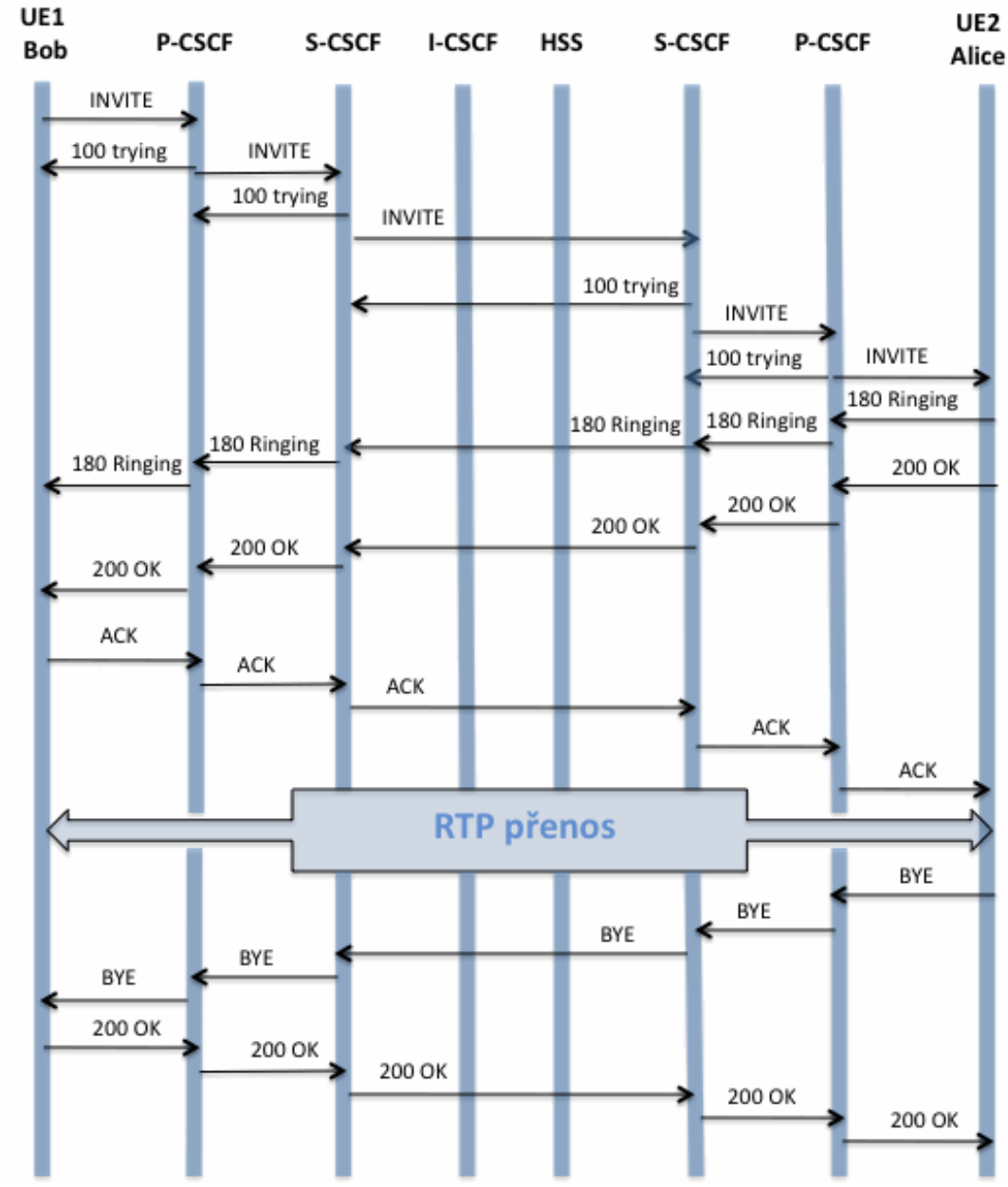
e) zpráva 200 OK z P-CSCF do UE2 (Alice)

Komunikace probíhá mezi portem 4060, tedy serverem P-CSCF a portem 5062, na kterém je uživatel UE2 (Alice).

```
|Time      | 127.0.0.1
|39.561    |         Status: 200 OK SIP: Status: 200 OK
|          |(4060) ------------------------------------> (5062)
```

Změny oproti předchozí hlavičce:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.0.1:5062;rport=5062;branch=z9hG4bK5862405872337de669668bc9f5
11d12b
```



Obr. 6-10 Ustálení a ukončení spojení

7 Laboratorní úloha - Open IMS Core

Cíl

Cílem této úlohy je seznámit studenty s architekturou a principem fungování sítí založených na technologii IMS (IP Multimedia Subsystem). Studenti budou prakticky seznámeni s prvky potřebnými k provozování IMS sítě a jejich základním nastavením. Dále si vyzkouší komunikaci mezi klienty této sítě a provedou analýzu zachycené komunikace.

Požadavky na pracoviště

- PC s OS Windows nebo Linux,
- SW pro práci s virtuálními stroji (VMware Player),
- ISO image se systémem Open IMS Core, postaveném na linuxové distribuci Gentoo s grafickým rozhraním KDE a paketovým analyzátorem Wireshark,
- SW klient pro komunikaci v IMS systému (OpenIC_Lite).

Úkoly

- Zprovoznění systému Open IMS Core - spuštění potřebných serverů.
- Komunikace mezi výchozími klienty v konzolovém režimu.
- Vytvoření nových účtů a jejich komunikace v OpenIC_Lite klientech.
- Zachycení registrace a komunikace mezi klienty ve Wiresharku.
- Analýza zachycené komunikace.

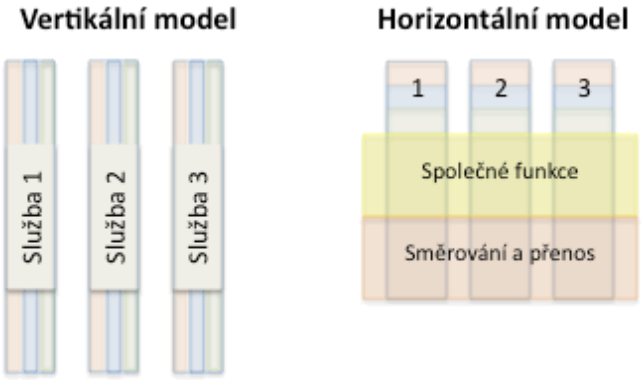
Teoretický úvod

V dnešní době již poskytovatelé telekomunikačních služeb implementovali do svých vybudovaných sítí většinu služeb, jenž ve svých sítích mohou provozovat s ohledem na přiměřené náklady. Nové služby, které by rádi poskytovali svým zákazníkům, si však vyžadují jiný způsob přenosu a zaměřují se zejména na využití IP protokolu (Internet Protocol). Implementace takových služeb však představuje nemalé investice do infrastruktury sítě poskytovatele telekomunikačních služeb. Řešením, které by co nejefektivněji implementovalo tyto služby, by mohl být systém zvaný IMS (IP Multimedia Subsystem).

IMS je tedy souhrn služeb spojených pomocí standardizovaných rozhraní, jenž má nahradit současné rozdělení na sítě s přepínáním okruhů (CS - Circuit Switched) a sítě s přepínáním paketů (PS - Packet Switched). Hlasové služby jsou dnes většinou realizovány přepínáním okruhů v CS doméně, kdežto přenos dat je realizován přepínáním paketů v PS doméně. Síť IMS sjednocuje přenos hlasu i dat do paketů, čímž zachovává současné schéma pro přenos dat a přenos hlasu převádí na VoIP platformu protokolu SIP, kde zavádí kvalitu řízení přenosu a priorizaci VoIP, tedy obdobu QoS (Quality of Service) v IP sítích.

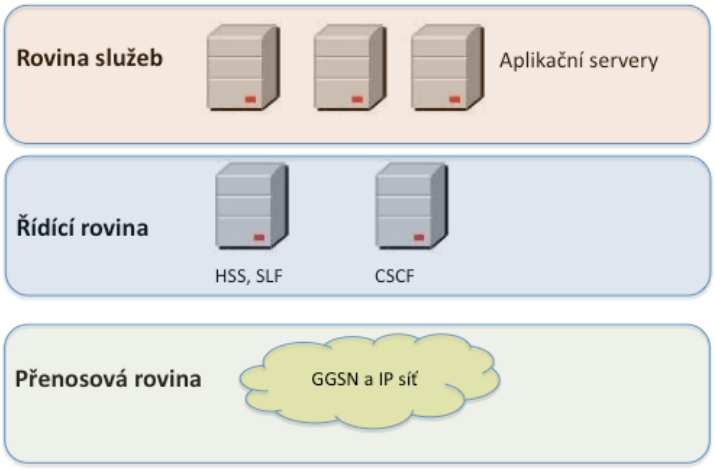
Doposud se všechny služby do mobilní sítě implementují složitým způsobem, při kterém je potřeba řešit spolupráci v několika rovinách - přenosové, řídící a v rovině služeb. To vyobrazuje tzv. vertikální model na Obr. 7-1. Přenosová rovina představuje

nejnižší úroveň, která se zabývá přenosem, směrováním a kódováním informace. Řídící rovina se zabývá sestavením, udržením a ukončením spojení. Rovina služeb představuje jak jednotlivé služby (aplikace), tak i zařízení, která si s těmito aplikacemi vyměňují informace (např. GPS lokalizační servery s informacemi o poloze uživatele).



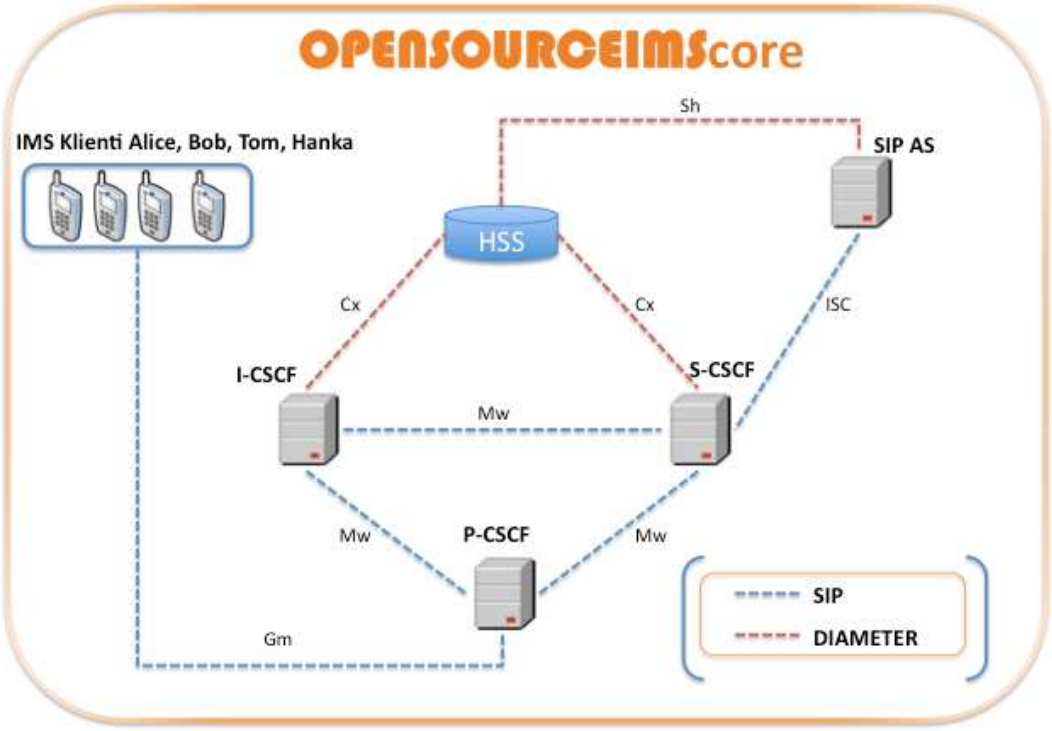
Obr. 7-1 Vertikální a horizontální model služeb

Právě IMS svou strukturou odpovídá horizontálnímu modelu, který je na Obr. 7-2. Výhoda takového modelu je jednodušší vývoj služeb a jejich snadná implementace. Pro správnou funkčnost tohoto modelu je nezbytné zajistit homogenitu přenosové a řídící roviny sítě. Toho se docílí právě sjednocením technologie, aby byla nezávislá na druhu přenášené informace.



Obr. 7-2 Horizontální model IMS

Open IMS Core představuje open source projekt Fraunhoferova Institutu FOKUS v Berlíně, jenž poskytuje základní implementaci pro testování IMS technologie a vývoj IMS aplikací. Je založen na linuxovém jádře, takže je jeho instalace možná jen na tyto systémy. Je tvořen SIP servery CSCF (I-CSCF, S-CSCF a P-CSCF) a jednoduchou databází uživatelů HSS, což je základ každé IMS architektury. Architekturu tohoto prostředí můžeme vidět na Obr. 7-3.



Obr. 7-3 Architektura systému Open IMS Core

Servery CSCF zde působí jako hlavní směrovací prvky pro jakoukoli IMS signalizaci. Jsou postaveny na SER (SIP Express Router), který může působit jako SIP registrátor, proxy nebo redirect server a je schopen zvládnout tisíce hovorů za vteřinu.

Jako databázi uživatelů HSS zde FOKUS vyvinul svou vlastní označovanou jako FHoSS (FOKUS Home Subscriber Server). Ta je kompletně napsána v jazyce Java a uživatelské informace jsou uloženy v databázi MySQL. Jelikož slouží jako databáze, tak se zaměřuje více než na výkon na pravdivost a shodu uložených informací.

Server HSS

Home Subscriber Server (HSS) je centrální úložiště pro informace týkající se uživatelů. Technicky se jedná o odvození z HLR (Home Location Server), který představuje uzel v síti GSM. HSS obsahuje všechna předplacená uživatelská data pro zpracování multimediálního spojení. Tyto data zahrnují kromě jiného informace o poloze, informace o zabezpečení (autorizace a autentizace), informace o profilu uživatele (obsahující předplacené služby uživatele) a server S-CSCF (Serving-CSCF) přidělený uživateli.

Servery CSCF

Server CSCF (Call/Session Control Function), jakožto SIP server, je nezbytný uzel v IMS. Zpracovává SIP signalizaci v IMS a jsou zde tři typy CSCF podle toho, jakou funkci provozují. Všechny se souhrnně označují jako CSCF a dále se dělí do těchto tří kategorií:

- P-CSCF (Proxy-CSCF),
- I-CSCF (Interrogating-CSCF),
- S-CSCF (Serving-CSCF).

Server **P-CSCF** je z pohledu signalizace prvním bodem kontaktu mezi IMS terminálem a IMS sítí. Z pohledu SIP vystupuje P-CSCF jako příchozí/odchozí SIP proxy server. To znamená, že všechny žádosti od IMS terminálu nebo žádosti směřující do IMS terminálu prochází přes P-CSCF. P-CSCF pak tyto SIP žádosti/odpovědi zasílá příslušnou cestou, a to směrem k IMS terminálu nebo k IMS síti.

P-CSCF je přidělen IMS terminálu během IMS registrace a po celou dobu registrace se nemění, takže IMS terminál po celou dobu registrace komunikuje jen s jedním P-CSCF.

Server **I-CSCF** je SIP proxy server umístěný na okraji administrativní domény. Adresa I-CSCF je uvedena v DNS (Domain Name System) záznamech. Kromě funkcí SIP proxy serveru má I-CSCF také rozhraní pro přístup k SLF a HSS. Tyto rozhraní jsou založeny na protokolu Diameter. I-CSCF obdrží informace o poloze uživatele a směruje SIP žádosti příslušnou cestou (typicky do S-CSCF).

Server **S-CSCF** představuje centrální bod na signalizační úrovni. Je to vlastně SIP server, ale provádí také řízení spojení. Kromě funkcí SIP serveru funguje S-CSFC také jako SIP registrátor. To znamená, že si udržuje spojení mezi lokací uživatele, např. IP adresu terminálu, ke kterému je uživatel přihlášen, a mezi uživatelovou adresou SIP záznamu - známou také jako veřejná identita uživatele - Public User Identity. Stejně jako u I-CSCF je i zde použito směrem k HSS rozhraní Diameter.

SIP AS

Jedná se o nativní aplikační server, který hostuje a vykonává IP multimediální služby založené na protokolu SIP.

Protokol SIP

Jedná se o textově orientovaný protokol, jenž se používá k signalizaci mezi dvěma více komunikujícími zařízeními v internetu, které si sám protokol je schopen vyhledat. Jeho předchůdcem v IP telefonii byl protokol H.323, který byl však velice složitý a protokol SIP se tak snaží být co nejjednodušší a vychází z protokolu HTTP.

Základní metody tohoto protokolu:

REGISTER - registrace,
INVITE - zahájení komunikace,
ACK - potvrzení zahájení relace,
CANCEL - přerušení zahajování relace,
BYE - ukončení relace.

Chybové hlášky protokolu:

1xx - průběh,
2xx - úspěch,
3xx - přesměrování,
4xx - chyba klienta,
5xx - chyba serveru,
6xx - fatální chyba.

Protokol Diamter

Jedná se o AAA protokol, tedy Authentication, Authorization and Accounting protocol řešící problémy autentizace autorizace a účtování. Jedná se o nástupce protokolu RADIUS a jako transportní vrstvu už používá spolehlivý TCP protokol narozdíl od UDP protokolu použitého u RADIUSu. Může použít zabezpečení na transportní vrstvě v podobě IPsec nebo TLS a poskytuje lepší podporu pro roaming.

Postup vypracování

I. Spuštění serverů

Na počítači si spustíme program VMware pro práci s virtuálním strojem a spustíme v něm připravený ISO image - OpenIMS. Po naběhnutí systému budou vyžadovány přihlašovací údaje - *Open IMS login: root, Password: password*. Nyní spustíme grafické rozhraní KDE pomocí následujícího příkazu: */etc/init.d/xdm start*, požadované přihlašovací údaje jsou stejné jako v předchozím kroku.

Otevřeme si terminálové okno, ve kterém nyní spustíme všechny potřebné servery pro práci v síti IMS. Spustíme jednotlivé servery CSCF a server s databází uživatelů HSS. Všechny tyto procesy by měly běžet paralelně a v jednotlivých oknech pak uvidíme pravidelný výpis logů těchto procesů. Tyto logy v případě výskytu problémů mohou blíže specifikovat vyskytlý problém.

```
cd /opt/OpenIMSCore
./pcscf.sh
./icscf.sh
./scscf.sh
cd /opt/OpenIMSCore/FHoSS/deploy
./startup.sh
```

Pro zastavení CSCF serverů slouží příkazy *killser CSCF* a *killall java*. Pro zastavení HSS pak příkaz *exit*. Pro usnadnění pohybu v adresářové struktuře můžeme použít utilitu *mc*.

II. Komunikace v konzolovém režimu

Nyní když máme spuštěny všechny potřebné servery, můžeme přistoupit ke spuštění klientů. Nyní použijeme předdefinované účty *Alice* a *Bob*, později si vytvoříme naše vlastní. Pro každého uživatele si otevřeme nové konzolové okno a spustíme jej následujícími příkazy:

```
cd /opt/OpenIMSCore
./Alice.sh (pro Boba analogicky ./Bob.sh)
```

Poté si příkazem "?" necháme vypsat přehled dostupných příkazů v konzolovém režimu a provedeme registraci těchto uživatelů. Po úspěšné registraci pak provedeme komunikaci mezi těmito uživateli zasláním libovolné zprávy. Všechny potřebné příkazy vyčteme z přehledu dostupných příkazů. Parametr URI pro zaslání zprávy má tvar: bob@open-ims.test.

III. Vytvoření nových uživatelů

Nyní si vytvoříme dva nové uživatele. Prvního pomocí konfiguračního skriptu, který se nachází v adresáři */opt/OpenIMSCore/ser_ims/cfg/*. Zde spustíme skript *add-imscore-user_newdb.sh* s parametrem *-u <user>* pomocí kterého definujeme nejen jméno nového uživatele, ale zároveň i jeho IMPI (soukromou identitu), IMPU (veřejnou identitu) a heslo. Druhým použitým parametrem pro spouštěný skript je *-a*, který nám zajistí automatické použití námi vytvořeného skriptu pro přidání nového uživatele *add-user-<user>.sql*.

Příklad pro přidání uživatele Tom:

```
cd /opt/OpenIMSCore/ser_ims/cfg/
./ add-imscore-user_newdb.sh -u tom -a
```

Druhého uživatele vytvoříme pomocí webového rozhraní k serveru HSS, a to na adrese http://localhost a portu 8080. Nejprve se musíme přihlásit pomocí následujících přihlašovacích údajů - login: *hssAdmin*, password: *hss*. Po vybrání záložky User Identities v horním menu se nám otevře v levé části okna nové menu s položkami IMS Subscription - Private Identity - Public User Identity. Z jednotlivých názvů je patrné, že se zde jedná o nastavení uživatelovy identity, jeho soukromé a veřejné identity. Pro vytvoření nového uživatele Tom (vy si zvolte odlišné) nejprve klikneme na položku Create v menu IMS Subscription. Zde je nutné vyplnit pole *Name: tom*, tedy jméno uživatele. Dále položku *Capabilities Set*, jenž představuje sadu přiřazených vlastností, vybereme zde *cap_set1* a nakonec preferovaný server S-CSCF v položce *Preferred S-CSCF*, zde vybereme *scscf1*.

Nyní vytvoříme soukromou identitu uživatele, kterou přiřadíme uživateli Tom. Vrátíme se opět do záložky *User Identities* a zde vybereme z menu *Private Identity* položku *Create*. Vyplníme zde pole *Identity: tom@open-ims.test*, tajný klíč *Secret Key: tom*, v položce Authentication Schemes si vybereme autentizační schéma, které chceme používat, např. Digest-AKAv1-MD5. Dále zde máme položky AMF - Authentication Management Field, jenž se využívá při autentizaci. V našem případě má délku 4B a nastavená hodnota *0000*. Položka OP - Operator ID, tedy ID operátora má 32B a má hodnotu *00000000000000000000000000000000*. Položka SQN - Sequence Number představuje sekvenční číslo o délce 12B s nastavenou hodnotou 000000000000.

Posledním krokem je z menu *User Identities* položka *Public User Identity*, jenž slouží pro nastavení veřejné identity uživatele. Zde zvolíme položku *Create* pro vytvoření nové veřejné identity. Vyplníme pole *Identity: sip:tom@open-ims.test*. Dále políčko Barring necháme odškrtnuté, jedná se o doplňkovou službu pro omezení volání např. příchozích nebo odchozích hovorů. Položka *Service Profile* představuje profil služeb a nastavíme ji hodnotu *deafult_sp*, nastavení spojené s informacemi o účtování necháme rovněž výchozí, takže pro položku *Charging-Info Set* vybereme *default_charging_set*. Zaškrtneme políčko *Can Register* pro možnost registrace. Pro typ veřejné identity možností *IMPU Type* vybereme *Public_User_Identity*, jelikož ta je určena pro IMS klienty. Položka *User-Status* nám poskytuje informaci o tom, zda je uživatel registrován či nikoliv.

IV. Komunikace v grafickém režimu

Nyní máme přichystané naše nové uživatele a můžeme je zaregistrovat do naší IMS sítě. K tomuto nyní využijeme klienta OpenIC_Lite, který se nachází v adresáři */opt/openIMSCore/OpenIC_Lite/*. Po spuštění nám klient najede do konfiguračního okna pro konfiguraci uživatelského účtu. Toto okno obsahuje tři záložky *User Profile*, *Server Profile* a *Application*. V *User Profile* definujeme zobrazované jméno pro uživatele (Tom), dále pak jeho veřejnou identitu (tom@open-ims.test), soukromou identitu (tom@open-ims.test) a tajný klíč (tom). V záložce *Server Profile* definujeme adresu proxy serveru (pcscf.open-ims.test), číslo portu (4060) a oblast pro rozřazení účastníků (open-ims.test). V poslední záložce *Application* máme možnost nastavit automatické přijímání hovorů, automatickou registraci klienta a možnost zapnout výstražná hlášení. Kompletní nastavení je pak nutné po souhlasu s licencí uložit tlačítkem *save*. Tímto se nám vytvoří konfigurační soubor *profile.cfg* v adresáři */opt/OpenIMSCore/OpenIC_Lite/*. Pokud jsme v předchozím kroku nezaškrtnuli políčko s automatickou registrací, tak

přejdeme do menu *File - Sign In*. Ve spodní části by se nám měl objevit text Registered, což nám oznamuje, že registrace tohoto účtu proběhla v pořádku. Toto si můžeme zkontrolovat i ve webovém rozhraní k hss na již známé adrese *http://localhost:8080*. Zde vidíme v menu *User Identities - IMS Subscription - Search*, že zaregistrovaným účtům je přiřazen S-CSCF name a Diameter Name. Pod položkou *Public User Identity - Search* pak máme u našeho účtu *Reg. Status = Registered*, čili registrován.

Pro následnou komunikaci si spustíme druhého vytvořeného uživatele v konzolovém režimu. Zde je nutné ještě vytvořit složku s profilem tohoto uživatele úpravou výchozích profilů v adresáři */opt/OpenIMSCore/OpenIC_Lite/*. Dále vytvoříme spouštěcí skript pro daného uživatele zkopírováním výchozího spouštěcího skriptu *Alice.sh* v adresáři */opt/OpenIMSCore/*, kde upravíme jen jeho název a řádek *cd OpenIC_Lite/Alice* podle jména našeho uživatele. V grafickém režimu je nejjednodušší volbou přidání si všech známých uživatelů do seznamu (*Phonebook*) v menu *Contact - Add Contact*. Zadáme povinné položky *Last Name*, *First Name* a *SIP URI* (tvar *sip:<adresa rozhraní>:<port>*). Poté již máme přístupné menu přes pravé tlačítko k zaslání zprávy (*Send Message*) či zahájení hovoru (*Call*). Opět provedeme zaslání libovolné zprávy pro ověření funkčnosti.

V. Zachytávání komunikace

Pro zachytávání komunikace nám poslouží předem nainstalovaný síťový analyzátor paketů Wireshark. Kromě Wiresharku lze pro analýzu komunikace v IMS síti použít také nástroje obsažené v samotném klientovi OpenIC_Lite, přístupné v menu *Tools - Diagnostics*. Zde se však jedná o zachytávání komunikace jen mezi IMS terminálem a P-CSCF. Také webové rozhraní HSS poskytuje jednoduchou analýzu přenášených zpráv a to právě mezi HSS a servery I-CSCF a S-CSCF. Zachytávání a následnou analýzu spustíme z menu *Statistic - Turn On Debug* . My si spustíme Wireshark a provedeme zachytávání paketů na rozhraní 127.0.0.1. Po té jednoho klienta odregistrujeme a provedeme znovu jeho registraci do IMS sítě. Po úspěšné registraci provedeme komunikaci mezi klienty a následně jejich odhlášení.

VI. Analýza přenášených zpráv

Zachycené pakety z předchozího bodu podrobte analýze. Zakreslete průběh registrace účastníka v IMS síti včetně příslušných serverů a jejich portů. Poté zakreslete průběh sestavení a ustálení spojení s následným přenosem zpráv. V přenášených paketech najděte vámi zaslané zprávy. Nezapomeňte filtrovat správné protokoly!

VII. Možný výskyt problémů

A)

Nastavení proměnné **java_home** u klienta - Při spuštění můžeme narazit na problém týkající se nastavení proměnné JAVA_HOME ve spouštěcím skriptu *OpenIC_Lite.sh*. Tato chyba může vypadat následovně:

```
OpenIMS ~ # mc
OpenIMS OpenIC_Lite # ./OpenIC_Lite.sh
./OpenIC_Lite.sh: line 20: /root/bin/java: No such file or
 directory
```

Je zde nutné zadat této proměnné cestu, kde se v našem systému nachází systémové prostředí JAVA. Pokud bychom tak neučinili, tak spouštěcí skript nebude schopen otevřít klienta, jenž ke svému spuštění využívá právě nástroje JAVA. Nastavíme tedy proměnné JAVA_HOME následující cestu (nebo jinou v závislosti na naší konkrétní konfiguraci):

```
JAVA_HOME: /usr/lib/jvm/sun-jdk-1.5/
```

B)

Chyba při spuštění serveru HSS hlásící již obsazený **port 8080**. Tato chyba může vypadat následovně:

```
Java.net.BindException: Address already in use:8080
```

K uvolnění portu 8080 musíme znát službu, která tento port obsadila. K tomuto nám poslouží příkaz ***netstat*** s příslušnými parametry, který nám vypíše název služby na daném portu spolu s jeho PID (Process ID). Poté příkazem ***kill*** tuto službu ukončíme, čímž dojde k uvolnění potřebného portu. Ukázka příkazu:

```
netstat -anp | grep 8080
kill <PID>
```

Kontrolní otázky

- Jaké jsou základní bloky IMS sítě a jejich funkce?
- Význam protokolu SIP a DIAMETER?
- Jaký je hlavní rozdíl mezi sítí IMS a současnými sítěmi s přepínáním okruhů a s přepínáním paketů?

Závěr

Cílem mé diplomové práce bylo prostudovat a popsat technologii IMS. Zpočátku jsem tedy popsal architekturu systému IMS, kde jsem zmínil hlavní prvky systému IMS kterými jsou: databáze HSS a SLF, SIP servery CSCF, aplikační servery AS, zdroj prostředků MRF, SIP server BGCF a brána PSTN. Jelikož projekt 3GPP nestandardizuje uzly nýbrž služby, můžeme tak o IMS říci, že je to souhrn služeb spojených pomocí standardizovaných rozhraní. Vývojáři tak mohou spojit dvě služby do jednoho uzlu nebo také rozdělit jednu službu mezi dva a více uzly.

Poté jsem se zaměřil na typy rozhraní, které se v IMS používají. Popsal jsem zde jednotlivá propojení různých částí sítí, použité protokoly a také jednotlivé zprávy, které se po těchto rozhraních přenáší. U rozhraní Cx a Sh jsem uvedl i přehled možných příkazů na tomto rozhraní. Mezi nejdůležitější rozhraní, které jsem zde popsal, patří rozhraní Gm, Mw, Cx, Dx a Sh.

Následně jsem popsal bezpečnostní procedury v IMS. Bezpečnost je v IMS rozdělena na zabezpečení přístupu (access security) a síťovou bezpečnost (network security). Zabezpečení přístupu zahrnuje autentizaci, což představuje ověření identity uživatele služeb, autorizaci, tedy přidělení oprávnění přístupu k určitým službám a také zabezpečení přenosu mezi IMS terminálem a sítí. Síťová bezpečnost se zabývá ochranou přenosu mezi síťovými uzly. Tyto uzly mohou, ale nemusí spadat pod stejného operátora. Popisuji zde také průběh počáteční registrace uživatele do IMS sítě.

Dalším úkolem bylo provézt realizaci IMS sítě v systému Open IMS Core. Tento systém je založen na linuxovém jádře, takže je jeho instalace možná jen na tyto systémy. Nejprve jsem tento systém nainstaloval na OS Ubuntu, kde však jednotlivé servery nedokázaly navázat komunikaci a nezavedly patřičné moduly. Z důvodu přetrvávajících problémů, které se nepodařilo vyřešit, jsem přistoupil k náhradnímu řešení, a to spuštění Open IMS Core pod OS Gentoo. Zde už se jednotlivé části sítě chovaly správně a síť se podařilo nakonfigurovat a provézt komunikaci mezi nově vytvořenými účastníky.

Po úspěšné konfiguraci IMS sítě jsem provedl analýzu přenášených zpráv. K tomuto jsem využil síťový analyzátor paketů Wireshark. Kromě Wiresharku jsem pro analýzu komunikace v IMS síti použil také nástroje obsažené v samotném IMS klientovi *OpenIC_Lite*. Zde se však jedná o zachytávání komunikace jen mezi IMS terminálem a serverem P-CSCF. Také webové rozhraní HSS poskytuje jednoduchou analýzu přenášených zpráv a to právě mezi databází HSS a servery I-CSCF a S-CSCF. V této analýze přenášené SIP signalizace jsem se zaměřil na počáteční registraci uživatele do IMS sítě a na sestavení, průběh a ukončení spojení mezi jednotlivými účastníky.

V závěru mé diplomové práce jsem vypracoval laboratorní úlohu, zaměřenou na práci v systému Open IMS Core. Cílem této úlohy je seznámit studenty s architekturou a principem fungování sítí založených na technologii IMS, praktické seznámení s jednotlivými prvky, nutnými pro provoz této sítě a jejich základním nastavením. Studenti si dále vyzkouší jednoduchou analýzu zachycené komunikace a získají tak ucelený přehled o architektuře IMS sítě a komunikaci v ní.

Seznam použité literatury

[1] POIKSELKA, Miikka, MAYER, Gregor, KHARTABIL, Hisham. The IMS: IP Multimedia Concepts and Services. England: WILEY, 2006. 431 s. Second edition. ISBN 0-470-01906-9.

[2] CAMARILLO, Gonzalo, GACÍA-MARTÍN, Miguel A. The 3G IP Multimedia Subsystem (IMS). England: WILEY, 2006. 427 s. Second edition. ISBN 0-470-01818-6.

[3] Technical Training Centre. Fundamentals of the IMS : IP multimedia subsystem. [s.l.] : [s.n.], 2007. 253 s.

[4] HORÁK, M. Vývoj služeb v mobilních sítích. Mobilní sítě [online]. 2008 [cit. 2008-12-01]. Dostupný z WWW: <http://access.feld.cvut.cz/rservice.php?akce=tisk&cisloclanku=2008030001>.

[5] WIKIPEDIE : otevřená encyklopedie [online]. Modified Wed, Sep 24, 2008 9:00:23 AM, [cit. 2008-11-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/IMS>.

[6] SIP Security and the IMS Core. Newport Networks [online]. 2008 [cit. 2008-12-03]. Dostupný z WWW: <http://www.newport-networks.com/whitepapers/SIPsecurity1.html>.

[7] IMS – IP Multimedia Subsystem : The value of using the IMS architecture. Ericsson : white paper [online]. 2004 [cit. 2008-11-17]. Dostupný z WWW: <http://www.scribd.com/doc/6633144/Ims-Ip-Multimedia-Subsystem>.

[8] IP Multimedia Subsystem (IMS). Sun microsystems : Telekomunikace [online]. 2008 [cit. 2008-10-15]. Dostupný z WWW: <http://cz.sun.com/solutions/tele/media.html>.

[9] IP Multimedia Subsystem . WIKIPEDIA : The Free Encyklopedia [online]. 2008 [cit. 2008-11-22]. Dostupný z WWW: <http://en.wikipedia.org/wiki/IP_Multimedia_Subsystem>.

[10] OPEN SOURCE IMS [online]. 2004 , Modified: Tue, Dec 9, 2008 9:57:52 AM [cit. 2008-12-12]. Dostupný z WWW: <http://www.openimscore.org/>.

[11] TOWLE, Thomas. IMS in Next Generation Networks : The Ip Multimedia Subsystem. Lucent Technologies : Bell Labs Innovations [online]. 2005 [cit. 2008-11-03]. Dostupný z WWW: <http://www.itu.int/ITU-T/worksem/ngntech/presentations /s1-towle.pdf>.

[12] REPIQUET, Joël . Tech-Invite [online]. 2005 , Last update: May 16, 2009 [cit. 2009-05-01]. Dostupný z WWW: <http://www.tech-invite.com>

[13] Fraunhofer FOKUS : OpenIC - The FOKUS Open IMS Client [online]. 2001 , Modified: Thu, May 21, 2009 [cit. 2009-04-15]. Dostupný z WWW: <http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/components/_openic/index.html>.

[14] SOUMAR, Michal. Signalizační protokol pro přenos hlasu přes datové sítě - SIP [online]. 7.1.2003 , Modified: Tue, Apr 7, 2009 [cit. 2009-04-26]. Dostupný z WWW: <http://www.elektrorevue.cz/clanky/03003/index.html>.

Seznam použitých zkratek

3GPP	The 3rd Generation Partnership Project
ADSL	Asymetric Digital Subscriber Line
AMR	Adaptive Multi-Rate compression
AS	Application Server
AUTN	Authentication token
AV	Authentication Vector
B2BUA	Back-to-Back User Agent
BGCF	Breakout Gateway Control Function
BICC	Bearer Independent Call Control
CAMEL AS	Customized Applications for Mobile network Enhanced Logic AS
CAP	CAMEL Application Part
CDF	Charging Data Function
CK	Cryptographic key
COPS	Common Open Policy Service
CRF	Charging Rules Function
CS	Circuit Switched
CS – MGCF	Circuit Switched – Media Gateway Control Function
CS-MGW	Circuit Switched - Media Gateway Function
CSCF	Call Session Control Functions
DNS	Domain Name Systém
EDGE	Enhanced Data GSM Environment
ESP	Encapsulated Security Payload
FHoSS	FOKUS Home Subscriber Server
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
gsmSCF	GSM Service Control Function
HLR	Home Location Server
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
I-CSCF	Interrogating – Call Session Control Function
ICID	IMS Charging IDentifier
ID	IDentifier
IK	Integrity Key
IKE	Internet Key Exchange
IM-SSF	IP Multimedia Service Switching Function
IMS	IP Multimedia Subsystem
IMS-ALG	IMS Application Layer Gateway
IMS-MGW	IMS - Media Gateway Function
IP	Internet Protocol
ISC	IMS Service Control
ISIM	IP Multimedia Services Identity Module
ISUP	ISDN User Part
LIR/LIA	Location-Info-Request/Answer
MAP	Mobile Application Part
MAR/MAA	Multimedia-Auth-Request/Answer
MGCF	Media Gateway Control Function

MGW	Media Gateway
MRF	Media Resource Functions
MRFC	Media Resource Functions Controllers
MRFP	Media Resource Functions Processors
MTP	Message Transfer Part
NAPTR	Name Authority Pointer
NAT-PT/NAPT-PT	Network Address Port Translator - Protocol Translator
OCS	Online Charging System
OSA	Open Service Architecture
OSA-SCS	Open Service Access-Service Capability Server
P-CSCF	Proxy – Call Session Control Function
PCM	Pulse Code Modulation
PDA	Personal Digital Assistant
PDF	Policy Decision Function
PDP	Packet Data Protocol, Policy Decision Point
PEP	Policy Enforcement Point
PLMN	Public Land Mobile Network
SCS	Service Capability Server
PNR/PNA	Push-Notification-Request/Answer
PPR/PPA	Push-Profile-Request/Answer
PS	Packet Switched, Presence server
PSI	Public Service Identity
PSTN	Public Switched Telephone Network
PUR/PUA	Profile-Update-Request/Answer, (PUA-Presence User Agent)
QoS	Quality od Service
R-UIM	Removable User Identity Module
RAND	Random challenge
RES	Response
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTR/RTA	Registration-Termination-Request/Answer
S-CSCF	Serving – Call Session Control Function
SA	Security Association
SAR/SAA	Server-Assignment-Request/Answer
SCS	Service Capability Server
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SEG	Security Gateway
SGW	Signaling Gateway
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SIP URI	SIP Uniform Resource Identifier
SLF	Subscription Locator Function
SNR/SNA	Subscribe-Notifications-Request/Answer
SPI	Security Parameter Index
SQN	Sequence Number
SRV	Service
SSF	Service Switching Function
TCP	Transport Control Protocol
THIG	Topology Hiding Inter-network Gateway

TLS	Transport Layer Security
TrGW	Transition Gateway
UAR/UAA	User-Authorization-Response/Answer
UDP	User Datagram Protocol
UDR/UDA	User-Data-Request/Answer
UE	User Equipment
UICC	Universal Mobile Telecommunications Systém
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
USIM	Universa Subscriber Identity Module
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
XRES	Expected response

Příloha A

Konfigurace koncových IMS účastníků pro konzolový režim

Tom - konfigurační soubor /opt/OpenIMSCore/OpenIC_Lite/Tom/profile.cfg

```
displayName=Tom
publicIdentities=sip:tom@open-ims.test
realm=open-ims.test
privateIdentity=tom@open-ims.test
secretKey=tom
proxyCSCF=pcscf.open.ims-test:4060/UDP
SQN=000000000000
AMF=0000
AMFSTAR=0000
OP=00000000000000000000000000000000
useAK=true
simulateISIM=false
sqnVectorCurrentIndex=0
IND_LEN=5
delta=268435456
L=32
```

Konfigurace koncových IMS účastníků pro grafický režim

Tom - konfigurační soubor /opt/OpenIMSCore/OpenIC_Lite/ profile.cfg

```
#Open IMS Client config file
OP=00000000000000000000000000000000
AMFSTAR=0000
secretKey=tom
showExitDialog=true
AMF=0000
autoPlaySound=true
SQN=000000000000
realm=open-ims.test
privateIdentity=tom@open-ims.test
proxyCSCF=pcscf.open.ims-test:4060/UDP
publicIdentities=sip\:tom@open-ims.test
displayName=Tom
useAK=true
earlyIMS=false
autoSignIn=false
autoAnswer=true
```

Spouštěcí skripty v adresáři /opt/OpenIMSCore

Ukázka spouštěcího skriptu pro konzolový režim:

Spuštěcí skript Tom.sh

```
#!/bin/bash
cd OpenIC_Lite/Tom
# ---------------------
# set JAVA_HOME to your own preferences
# ---------------------
#JAVA_HOME=/usr/lib/java
# ------------------------------------------------------------------
# Include JAR Files
# ------------------------------------------------------------------
CLASSPATH=$CLASSPATH
for i in ../lib/*.jar; do CLASSPATH="$i":"$CLASSPATH"; done
# ------------------------------------------------------------------
# Start-up
# ------------------------------------------------------------------
$JAVA_HOME/bin/java -cp $CLASSPATH -Djava.library.path=lib
de.fhg.fokus.ims.IMSConsoleUE
cd ../..
```

Pro grafické rozhraní by tento skript vypadal takto:

Spouštěcí skript Tom_GUI.sh

```
#!/bin/bash
cd OpenIC_Lite/Tom
# ---------------------
# set JAVA_HOME to your own preferences
# ---------------------
JAVA_HOME=/usr/lib/jvm/sun-jdk-1.5/
# ------------------------------------------------------------------
# Include JAR Files
# ------------------------------------------------------------------
CLASSPATH=$CLASSPATH
for i in ../lib/*.jar; do CLASSPATH="$i":"$CLASSPATH"; done
# ------------------------------------------------------------------
# Start-up
# ------------------------------------------------------------------
$JAVA_HOME/bin/java -cp $CLASSPATH -Djava.library.path=../lib
OpenIMSDemoClient
cd ..
```

Příloha B

Ukázka skriptu pro přidání nového uživatele "Hanka" : /opt/openIMSCore/ser_ims/cfg/add-user-hanka.sql

```
insert into hss_db.imsu(name) values ('hanka_imsu');
#add Private Identity
#Add hanka@open-ims.test
insert into hss_db.impi( identity, id_imsu, k, auth_scheme,
                         default_auth_scheme, amf, op)
values( 'hanka@open-ims.test',
        (select       id       from       hss_db.imsu       where
hss_db.imsu.name='hanka_imsu'),
        'hanka',
        127,
        1,
        '\0\0',
        '\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0');

#--add Public SIP Identity
insert into hss_db.impu(identity,id_sp) values ('sip:hanka@open-
ims.test', (select id from hss_db.sp order by id limit 1));
update     hss_db.impu     set     id_implicit_set=id     where
hss_db.impu.identity='sip:hanka@open-ims.test';

#--add Public Identity to Private Identity
insert into hss_db.impi_impu(id_impi,id_impu) values ((select id
from    hss_db.impi    where    hss_db.impi.identity='hanka@open-
ims.test'),     (select     id     from     hss_db.impu     where
hss_db.impu.identity='sip:hanka@open-ims.test'));

#--add roaming network
insert          into          hss_db.impu_visited_network(id_impu,
id_visited_network)  values((select  id  from  hss_db.impu  where
hss_db.impu.identity='sip:hanka@open-ims.test'), (select id from
hss_db.visited_network                                          where
hss_db.visited_network.identity='open-ims.test'));
```

Skript pro smazání uživatele "Hanka" : /opt/openIMSCore/ser_ims/cfg/delete-user-hanka.sql

```
delete from hss_db.impu_visited_network where id_impu = (select
id from hss_db.impu where hss_db.impu.identity='sip:hanka@open-
ims.test');
delete from hss_db.impi_impu where id_impi = (select id from
hss_db.impi where hss_db.impi.identity='hanka@open-ims.test');
delete  from  hss_db.impu  where  identity  =  'sip:hanka@open-
ims.test';
delete from hss_db.imsu where name = 'hanka_imsu';
delete from hss_db.impi where identity = 'hanka@open-ims.test';
```

Příloha C

Registrace uživatele Tom - zachycení zpráv pomocí Open IC Lite

```
Sat May 09 17:50:52 CEST 2009
SENT REQUEST>> REGISTER sip:open-ims.test SIP/2.0
Call-ID: 36fd142548ccfc19e27c555279587f5e@127.0.0.1
CSeq: 1 REGISTER
From: "Tom" <sip:tom@open-ims.test>;tag=1000
To: "Tom" <sip:tom@open-ims.test>
Via: SIP/2.0/UDP
127.0.0.1:5063;branch=z9hG4bK65cd4ea41740cf2dbcb408dbfbc38dc0
Max-Forwards: 20
Expires: 3600
Authorization: Digest username="tom@open-ims.test",realm="open-
ims.test",nonce="",response="",uri="sip:open-ims.test"
Supported: path
Contact: <sip:127.0.0.1:5063>
P-Preferred-Identity: "Tom" <sip:tom@open-ims.test>
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-
3gpp=00000000
Privacy: none
User-Agent: Fraunhofer FOKUS/NGNI Java IMS UserEndpoint FoJIE 0.1
(jdk1.3)
Allow: INVITE,ACK,CANCEL,BYE,MESSAGE,NOTIFY
Content-Length: 0


Sat May 09 17:50:53 CEST 2009
INCOMING RESPONSE >> SIP/2.0 401 Unauthorized - Challenging the
UE
Call-ID: 36fd142548ccfc19e27c555279587f5e@127.0.0.1
CSeq: 1 REGISTER
From: "Tom" <sip:tom@open-ims.test>;tag=1000
To: "Tom" <sip:tom@open-
ims.test>;tag=d7837ce6bbd631122d10546eb75bb4cf-3782
Via: SIP/2.0/UDP
127.0.0.1:5063;rport=5063;branch=z9hG4bK65cd4ea41740cf2dbcb408dbf
bc38dc0
Path: <sip:term@pcscf.open-ims.test:4060;lr>
Service-Route: <sip:orig@scscf.open-ims.test:6060;lr>
Allow:
INVITE,ACK,CANCEL,OPTIONS,BYE,REFER,SUBSCRIBE,NOTIFY,PUBLISH,MESS
AGE,INFO
Server: Sip EXpress router (2.1.0-dev1 OpenIMSCore (i386/linux))
Warning: 392 127.0.0.1:6060 "Noisy feedback tells:  pid=9478
req_src_ip=127.0.0.1 req_src_port=5060 in_uri=sip:scscf.open-
ims.test:6060 out_uri=sip:scscf.open-ims.test:6060 via_cnt==3"
WWW-Authenticate: Digest realm="open-
ims.test",nonce="98f9e2d0ad88818f0ca5b8b10d0f2086",algorithm=MD5,
qop="auth,auth-int"
Content-Length: 0
```

```
Sat May 09 17:50:53 CEST 2009
SENT REQUEST>> REGISTER sip:open-ims.test SIP/2.0
Call-ID: 36fd142548ccfc19e27c555279587f5e@127.0.0.1
CSeq: 2 REGISTER
From: "Tom" <sip:tom@open-ims.test>;tag=1001
To: "Tom" <sip:tom@open-ims.test>
Via: SIP/2.0/UDP
127.0.0.1:5063;branch=z9hG4bK731084d1ab1013b5ad8109b0948d47d7
Max-Forwards: 20
Contact: <sip:127.0.0.1:5063>
Expires: 3600
Authorization: Digest username="tom@open-ims.test",realm="open-
ims.test",nonce="98f9e2d0ad88818f0ca5b8b10d0f2086",uri="sip:open-
ims.test",algorithm=MD5,response="820deb3e6ebfce4bf0e16f46e0f8128
6",qop=auth-int,nc=00000001,cnonce="102565348491005655"
Supported: path
P-Preferred-Identity: "Tom" <sip:tom@open-ims.test>
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-
3gpp=00000000
Privacy: none
User-Agent: Fraunhofer FOKUS/NGNI Java IMS UserEndpoint FoJIE 0.1
(jdk1.3)
Allow: INVITE,ACK,CANCEL,BYE,MESSAGE,NOTIFY
Content-Length: 0


Sat May 09 17:50:53 CEST 2009
INCOMING RESPONSE >> SIP/2.0 200 OK - SAR succesful and registrar
saved
Call-ID: 36fd142548ccfc19e27c555279587f5e@127.0.0.1
CSeq: 2 REGISTER
From: "Tom" <sip:tom@open-ims.test>;tag=1001
To: "Tom" <sip:tom@open-
ims.test>;tag=d7837ce6bbd631122d10546eb75bb4cf-d59a
Via: SIP/2.0/UDP
127.0.0.1:5063;rport=5063;branch=z9hG4bK731084d1ab1013b5ad8109b09
48d47d7
P-Associated-URI: <sip:tom@open-ims.test>
Contact: <sip:127.0.0.1:5063>;expires=3600
Path: <sip:term@pcscf.open-ims.test:4060;lr>
Service-Route: <sip:orig@scscf.open-ims.test:6060;lr>
Allow:
INVITE,ACK,CANCEL,OPTIONS,BYE,REFER,SUBSCRIBE,NOTIFY,PUBLISH,MESS
AGE,INFO
P-Charging-Function-Addresses: ccf=pri_ccf_address
Server: Sip EXpress router (2.1.0-dev1 OpenIMSCore (i386/linux))
Warning: 392 127.0.0.1:6060 "Noisy feedback tells:  pid=9479
req_src_ip=127.0.0.1 req_src_port=5060 in_uri=sip:scscf.open-
ims.test:6060 out_uri=sip:scscf.open-ims.test:6060 via_cnt==3"
Content-Length: 0
```